AD-A088 899 AIR FORCE WEAPONS LAB KIRTLAND AFB NM F/G 18/3
A USERS' MANUAL FOR EXTENSIONS TO THE NOVA-2 AND NOVA-2S COMPUT--ETC(U)
APR 80 G M CAMPBELL
UNCLASSIFIED AFWL-TR-79-160 SBIE-E200 544 NL

END
DATE
FILMED
10-80
DTIC

AP-E200544

AFWL-TR-79-160 (2) LEVEL III

AFWL-TR-79-160

AD A088899

A USERS' MANUAL FOR EXTENSIONS TO THE NOVA-2 AND NOVA-2S COMPUTER CODES

April 1980

Gerald M. Campbell

Final Report

Approved for public release; distribution unlimited.

80 6 6 135

AIR FORCE WEAPONS LABORATORY
Air Force Systems Command
Kirtland Air Force Base, NM 87117

AFWL-TR-79-160

This final report was prepared by the Air Force Weapons Laboratory, Kirtland Air Force Base, New Mexico, under Job Order 88090326. Mr. Gerald M. Campbell (NTYV) was the Laboratory Project Officer-in-Charge.

This technical report has been reviewed and is approved for publication.

Gerald M. Campbell

GERALD M. CAMPBELL
Project Officer

FOR THE DIRECTOR

Terry A. Schmidt

TERRY A. SCHMIDT
Major, USAF
Chief, Environment and Effects Branch

Thomas W. Ciambrone

THOMAS W. CIAMBRONE
Colonel, USAF
Chief, Applied Physics Division

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFWL-TR-79-160	2. GOVT ACCESSION NO. AD-A088 899	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A USERS' MANUAL FOR EXTENSIONS TO THE NOVA-2 AND NOVA-2S COMPUTER CODES		5. TYPE OF REPORT & PERIOD COVERED Final Report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Gerald M. Campbell		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Weapons Laboratory (NTYV) Kirtland Air Force Base, NM 87117		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62601F/88090326
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Weapons Laboratory (NTYV) Kirtland Air Force Base, NM 87117		12. REPORT DATE April 1980
		13. NUMBER OF PAGES 66
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Aircraft Vulnerability
Digital Computer Program
Structural Response Analysis
Stiffened Panels
Elastic-Plastic Material
Nuclear Blast
Overpressure Effects

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The NOVA-2 and NOVA-2S computer codes have been extended to provide the option for specifying arbitrary dynamic pressure loads on beam and panel structural elements. An option has been included to allow the user to input digitized pressure data (obtained from experimental tests) from magnetic tape. Another modification provides the option of pinned boundary conditions in rib buckling analyses. A free boundary condition and discrete linear elastic spring option

(over)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. ABSTRACT (Cont'd)

have been included in the stiffened panel version, NOVA-2S. A technique for using radial imperfections to approximate the curvature of a noncylindrical panel is discussed.

PREFACE

This report covers extensions to the NOVA-2 and NOVA-2S structural dynamic response computer codes documented in AFWL-TR-75-262 and AFWL-TR-78-182. The basis for the current report was obtained from previous work performed by Kaman AviDyne under contract to the Air Force Weapons Laboratory and the Defense Nuclear Agency.

CONTENTS

ILLUSTRATIONS

TABLES

I. INTRODUCTION

The NOVA-2 (Nuclear Overpressure Vulnerability Analysis, Version 2) computer code, documented in Reference 1, was developed for use in analyzing the dynamic response of aircraft structure to transient pressure loads associated with the blast wave from a nuclear explosion. Since then a number of changes have been made to the code to increase its versatility. The purpose of this report is to compile these changes so that they will be more readily available to the user.

The first major change to the NOVA-2 code (NOVA-2L, Refs. 2 and 3) was made to provide the analyst with a versatile set of pressure loading routines so analytical predictions could be compared with experimental tests such as those conducted in the Sandia Laboratories' Thunderpipe shock tube facility (Ref. 4).

The pressure loading routines in the NOVA-2L code are analytical in form and require a certain amount of manual processing of the test data. In order to eliminate most of this time-consuming work, changes were made to allow the analyst to input digitized experimental pressure data directly from magnetic tape, and the NOVA-2LT code (Ref. 5) was created. All of the pressure load options from NOVA-2L are contained in NOVA-2LT unchanged. The changes necessary to transform NOVA-2 to NOVA-2LT are listed in Appendix A.

Similar changes are documented in Reference 6 for the NOVA-2S code, a stiffened panel extension to the basic NOVA-2 code. This special version is called NOVA-2LTS where LT refers to the fact that all of the nuclear blast and aeronautical loading routines have been replaced by user-generated functions or digitized test data on tape.

Additional changes were made to the NOVA-2 (NOVA-2LT) and NOVA-2S (NOVA-2LTS) versions to include a pinned/sliding-pinned end constraint. NOVA-2LTS was also revised to include free-free and clamped-free edge conditions, and elastic springs along the edges (Ref. 7). These changes are documented in Appendixes B and C.

Section II describes the pressure load options available to specify the loading applied to the structural element model.

Section III covers program description and operation, including rib element end constraints, stiffened panel edge fixity, and input data instructions.

Section IV describes a method for approximating the curvature of a noncylindrical panel using geometric radial imperfections.

II. DYNAMIC LOAD OPTIONS

Four transient pressure models and one static pressure model are incorporated into the NOVA-2LT and NOVA-2LTS codes. Two of the transient models are uniform spatially, i.e., the pressure functions vary with time only. The static pressure model, of course, is independent of time and is represented by specifying either a positive or negative pressure, P_s, to represent the direction and magnitude of the loading. This is consistent with NOVA-2, where a positive pressure is taken to be acting inward. A combination static-dynamic response calculation can be made where the static solution is determined first, and then followed by the dynamic load and resulting response.

For cases where the code is being used in the iterative modes, range is replaced by a range factor parameter which is normalized to 1.0 initially. For subsequent trials this parameter acts like a range; but it is actually the inverse of the factor which adjusts the pressure loading. Only the magnitude of the pressures is affected, not the characteristic times.

1. DYNAMIC LOAD OPTION 1--SPATIALLY UNIFORM LOAD WITH AN ANALYTICAL DECAY FUNCTION.

The first dynamic load option consists of a uniform pressure (P_1) with an analytical decay function designed to approximate the diffraction and drag phases of a blast loading. This function (see Figure 1) represents a combination linear and exponential decay:

$$P_1\ (t) = P_1\left(1 - \frac{t}{t_1}\right) \qquad (t < t')$$

$$P_{II}(t) = P_o\left(1 - \frac{t}{t_o}\right)^n e^{-at/t_o} \quad (t' \leq t < t_o)$$

$$P_{III}(t) = 0 \qquad\qquad (t \leq t_o) \qquad\qquad (1)$$

The second function of Equation 1 is used for time greater than or equal to t'. By specifying t' = 0, the special loading cases indicated in Table 1 can easily be generated, where I is the impulse and Δt is the integration time interval.

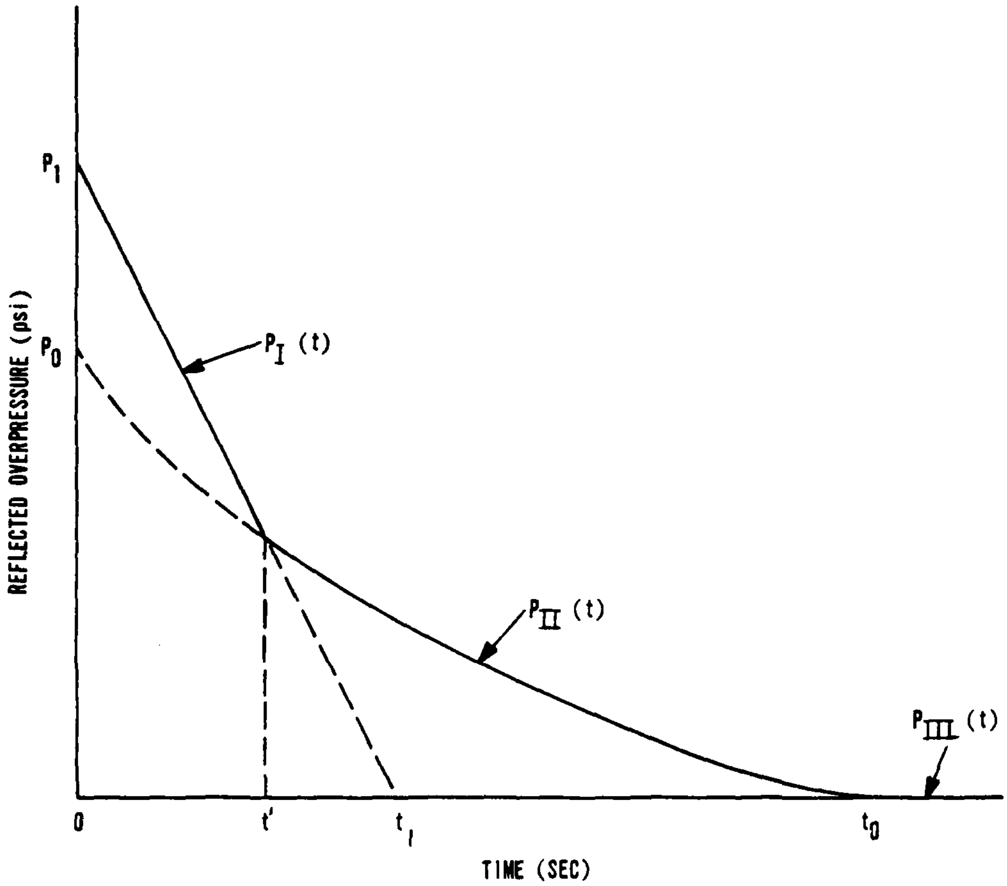

Figure 1. Analytical pressure time history

TABLE 1. LOADING PARAMETERS FOR SPECIAL CASES

Parameter	Step Function	Triangular Decay	Impulse	Exponential Decay
P_1	0	0	0	0
P_o	P_o	P_o	$\frac{2I}{\Delta t}$	P_o
t_o	1.0×10^{10}	t_o	1.0×10^{-20}	t_o
t'	0	0	0	0
a	0	0	0	a
n	0	1	1	0

2. DYNAMIC LOAD OPTION 2--SPATIALLY UNIFORM DISCRETE LOADS.

Like the previous option, loading option 2 is appropriate for uniformly applied loads without consideration of engulfment. With this option, discrete values of pressures can be specified at a set of times beginning at zero. For other times, linear interpolation is used except after the last time in the table, where a pressure equal to the last value is assumed.

3. DYNAMIC LOAD OPTION 3--SPATIALLY NONUNIFORM DISCRETE LOADS.

The third load option permits a spatially nonuniform, discrete load description. As in option 2 the transient load is specified at user-selected times.

For beam element analysis, the spatial variation is accomplished by initially selecting pressure measurement locations along the beam element. These positions are indicated in terms of mass point positions, i.e., position 3.67 would indicate a measurement position 2/3 of the distance between mass numbers 3 and 4 in the beam element model. Each measurement station then has its own time history.

A rectangular mesh of measurement positions must be used for two-dimensional, panel elements. This mesh need not be the same as the rectangular integration grid used in determining the structural response. Measurement positions are specified in terms of x, y coordinates. Again, each mesh point station has its own unique time history.

Interpolations for mass points or integration points located between measurement stations are made linearly. Linear extrapolation is used for points outside the mesh.

Linear interpolation is also used between break points in the temporal description except for times greater than the last table point, where a pressure equal to the last value is assumed. At least one point in the field must be intercepted at time zero.

4. DYNAMIC LOAD OPTION 4--EXPERIMENTAL TEST PRESSURE--TIME HISTORY FROM DIGITIZED DATA TAPE.

The fourth load option provides for user supplied pressure-time histories on magnetic tape. After the experimental test data have been filtered and digitized, it is stored on data tapes in a specific format which is discussed in Section III. Each tape contains multiple files where each file corresponds to a particular pressure gage. This, of course, precludes the possibility of iterating for critical response, because only one set of pressure data is available. The data tape is mounted prior to program execution, and the data are transferred to large core memory (available on computers such as the Control Data Corporation 7600 and CYBER 176) before the structural response analysis begins.

During execution both temporal and spatial variation in pressure are accomplished through linear interpolation, though the spatial variation is limited to only one coordinate direction. This means that for panels the variation can either be circumferential or axial, but not both. In case of a uniform spatial distribution, only one pressure time history is specified and spatial interpolation is not attempted. The temporal variation is limited to the extent that the times must be regular intervals apart and the total number of samplings be within the dimensional limits in the program.

The measurement positions are specified in the same manner as described in Section II.3. They must be coordinated with the tape data on a gage-to-gage basis so that the right data get applied to the appropriate place on the beam or panel. One set of input data specifies which tape file corresponds to which measurement station. It is also possible to exclude an entire set of data if that gage is later found to have been defective. The measurement stations are assumed to be ordered consecutively from the beginning of the beam model, or from the coordinate origin for panel models.

In general, the time history contained on tape will include times prior to the shock arrival at the gage nearest the blast. Therefore, it is necessary to scan a plot of the pressure-time history data obtained from that gage to determine the start time for input of pressures into the NOVA-2LT(S) code.

The program also provides for skipping over portions of the pressure-time history. The input parameter (SKIP) indicates the frequency at which the tape data is sampled. For example, SKIP=3.0 would mean that every third time point following shock arrival (at the nearest gage) will be used to describe the time history. Using less data should mean a somewhat faster running program, but this also sacrifices some accuracy. It is the user's responsibility to determine how much accuracy is required for the pressure-time history. It is also important that the program integration stop time (TSTOP) selected does not exceed the data available on the tape.

III. PROGRAM DESCRIPTION AND OPERATION

The NOVA-2LT and NOVA-2LTS codes represent simplified versions of NOVA-2 since 52 subroutines have been eliminated. However, loader segmentation is still used to optimize computer core requirements and flexibility. If segmentation is not used, the user must select the necessary routines from Table 2, depending on which subprogram is being exercised. The user should realize that large core memory (LCM) is required when using experimental test data tapes (load option 4), thus restricting code use to computers such as the Control Data Corp 7600 and CYBER 176 systems.

The amount of central memory required can be reduced by separating the code (NOVA-2LT or NOVA-2LTS) into two segmented codes: a beam element code (DEPROB) and a panel element code (DEPROP).

1. BEAM AND PANEL SUBPROGRAMS.

Since the NOVA-2 and NOVA-2S program libraries are maintained at the Air Force Weapons Laboratory (AFWL) using the Control Data Corp 6000/7000 UPDATE program, it is relatively simple to assemble only the subroutines needed to create the desired code (DEPROB or DEPROP) along with the changes necessary to transform the basic codes to the specialized LT versions discussed in Section II. These subroutines are categorized in Table 2. Note that certain subroutines are common to both codes.

2. RIB BUCKLING CHANGES FOR PINNED BOUNDARY CONDITIONS.

The NOVA-2LT (NOVA-2LTS) codes have been modified to include pinned/sliding-pinned end constraints for the rib option (KTYPE=10). Provision is also made for including the mass of the piston and loading block in the analytical model to simulate the test columns in the Boeing Phase II STRESNO program (Ref. 4).

The rib buckling changes, documented in Appendix B, include a statement, "SM(1)=XXX/DELTS(1)", where "XXX" represents the mass in lb-s^2/in. Since this first mass point actually represents a piston and loading block in the model, the link between the first end constraint at boundary coordinates (V1, W1) and the first mass point at coordinates [V(1), W(1)] is a pseudo link. Initially, this link must be horizontal as shown in Figure 2.

TABLE 2. SUBPROGRAMS FOR NOVA-2LT (NOVA-2LTS)

COMMON SUBROUTINES	DEPROP	DEPROB
NOVA	DEPROP	DEPROB
SEC	BOLT	COMP1
RITER	[a]DERV1	COMP2
CSETUP	DERV2	COMSET
PINIT	DSET1	CYCLE
SOLVE	DSET2	DAB
INTL	DSET3	DEFORM
PRESS	DTSTEP	DPUR
	HIM	EQUILP
	LEGEND	EQUILX
	LIST1	FB
	LIST2	FBCTL
	[a]MATXIN	FBSET
	RELAXP	FINAL
	SIGMA	FSOL
	[a]SIGMAB	PRINT1
	[a]STIFF	READ1
		RESD
		RESET
		RLAXB
		RLAXF
		SLAY
		STRESS
		STRESX
		STRN1
		STRN2
		STSET
		TSTEP
		VCS

[a]Required for stiffened panel version only.

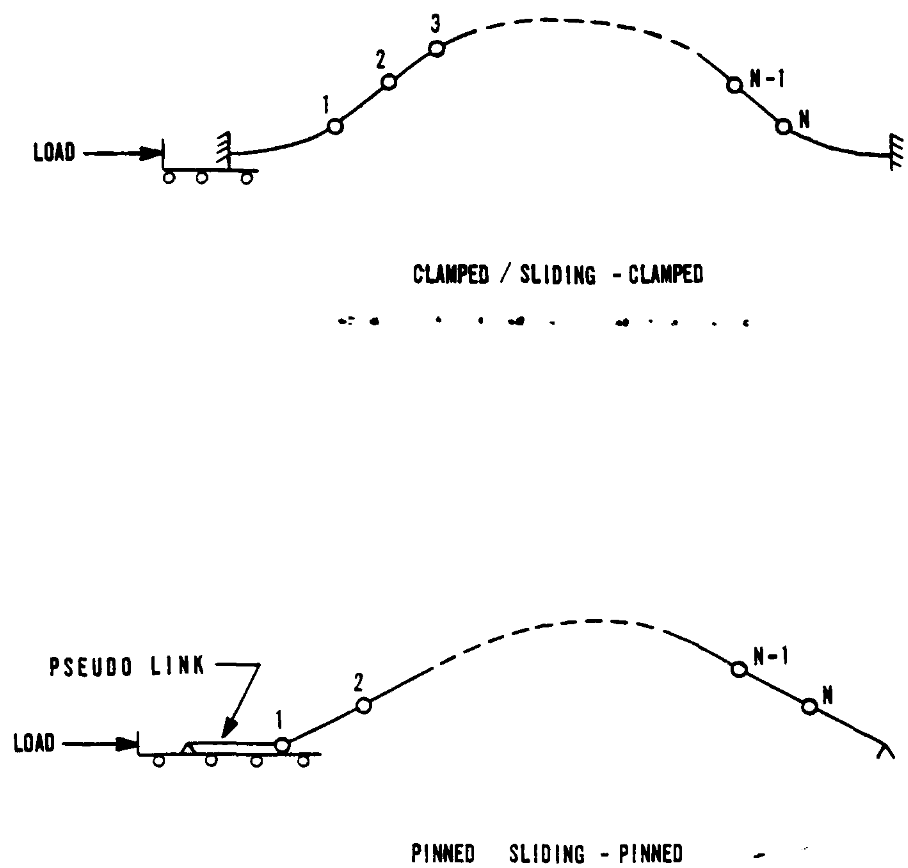

Figure 2. Rib buckling model.

Initial geometric imperfections (in the w direction) are required in the rib to induce buckling. If the default option is used, a 1-cosine mode shape is applied to the clamped case, and a sine mode shape to the pinned case. When exercising this option, the input parameter AMP (Group 21, p. 277, Ref. 1) represents the magnitude of the initial imperfection at the center of the rib. The other method of specifying an imperfection utilizes the V(I), W(I) mass point coordinate of the rib (Group 6, p. 274 of Ref. 1).

3. STIFFENED PANEL CHANGES FOR FREE BOUNDARY CONDITIONS AND DISCRETE LINEAR SPRINGS.

The NOVA-2S (NOVA-2LTS) codes have been modified to allow panels to be modeled with free edge boundaries and discrete linear springs at arbitrary positions. These options were added to allow the user to analyze structures such as the center bomb bay doors of the B-52 aircraft. The center door is nearly clamped along one edge and free along the other three edges except at the two corners which connect to the adjacent doors. These corner connections can be simulated by springs representing the compliance of the adjacent doors. The springs must be located (in any of the three orthogonal coordinate directions) at the spatial integration points γ_j, β_k defined for the unstiffened panel either along the boundary edges or within the panel's interior. It is assumed that the coordinate direction of the spring forces are deformation dependent, that is, the spring force always remains directed radially (w), tangentially (v), or axially (u) as the panel deforms. Thus, the following expression is added to the equaton of motion in Equation 13 of Reference 6:

$$\frac{2\tau^2 L^2 R}{\theta_o \ell} \sum_{i=1}^{S_w} K_i^w \, W(\gamma_j, \beta_k) \frac{\partial W(\gamma_j \beta_k)}{\partial W_{mn}} \qquad (2)$$

where

K_i^w = spring stiffness in w-direction (lbs/in)

S_w = total number of springs in w-direction

The other nomenclature is defined in References 1 and 6. Similar additions have been made for the u and v equations of motion using K^u and K^v over S_u and S_v, respectively. In the future, rotational springs should be included in a similar manner, so that boundary conditions between clamped and simply supported can be simulated.

The boundary combinations for the γ and β directions for the w-displacement have been extended to include the free-free and clamped-free conditions. The w-displacement functions for the free-free condition in the γ and β directions are given by

$$\phi^w_{m=1} = 1, \quad \phi^w_{m=2} = \sqrt{3}\left(1 - \frac{2\gamma}{\pi}\right)$$

$$\phi^w_m = \cosh\frac{\lambda_m\gamma}{\pi} + \cos\frac{\lambda_m\gamma}{\pi} - \alpha_m \sinh\left(\frac{\lambda_m\gamma}{\pi} + \sin\frac{\lambda_m\gamma}{\pi}\right)$$

$$m = 3, 4, 5, \ldots$$

$$\phi^w_{n=1} = 1, \quad \phi^w_{n=2} = \sqrt{3}\left(1 - \frac{2\beta}{\pi}\right) \tag{3}$$

$$\phi^w_n = \cosh\frac{\lambda_n\beta}{\pi} + \cos\frac{\gamma_n\beta}{\pi} - \alpha_n\left(\sinh\frac{\gamma_n\beta}{\pi} + \sin\frac{\gamma_n\beta}{\pi}\right)$$

$$n = 3, 4, 5 \ldots$$

where

λ = the roots of $\cos\lambda \cosh\lambda = 1$

$$\alpha = \frac{\cosh\lambda - \cos\lambda}{\sinh\lambda - \sin\lambda}$$

The w-displacement functions for the clamped-free condition in the λ and β directions are given by

$$\phi^w_m = \cosh\frac{\lambda_m\gamma}{\pi} - \cos\frac{\lambda_m\gamma}{\pi} - \alpha_m\left(\sinh\frac{\lambda_m\gamma}{\pi} - \sin\frac{\lambda_m\gamma}{\pi}\right)$$

$$m = 1, 2, 3, \ldots \tag{4}$$

$$\phi^w_n = \cosh\frac{\lambda_n\beta}{\pi} - \cos\frac{\lambda_n\beta}{\pi} - \alpha_n\left(\sinh\frac{\lambda_n\beta}{\pi} - \sin\frac{\lambda_n\beta}{\pi}\right)$$

$$n = 1, 2, 3, \ldots$$

where

λ = the roots of $\cos\lambda \cosh\lambda = -1$

$$\alpha = \frac{\cosh\lambda + \cos\lambda}{\sinh\lambda + \sin\lambda}$$

For flat panels, these changes produce a good approximation of the actual boundary condition, but the bomb bay doors are curved and additional boundary conditions in the u and v directions are required. Presently, the u and v

boundary conditions represent a held inplane condition along all edges. Held-free and free-free boundary conditions are needed and should be included in future modificatons to the code.

The changes required to extend NOVA-2S (NOVA-2SLTS) to include free-free and clamped-free edge conditions and elastic springs are presented in Appendix C.

4. PROGRAM SEGMENTATION AND OPERATION.

The recommended segmentation tree structure is presented in Figure 3 for DEPROP, and in Figure 4 for DEPROB. The common blocks, which must be designated GLOBAL and SAVE in the segmentation loader directives, are listed in Table 3 along with the routines to which they are assigned. The subroutines SIGMA and SIGMAB, and the associated common block CBLANK are required only for the elastic-plastic response option of DEPROP. This option corresponds to selecting NDERV=2 in Group 5, page 115 of Reference 6. Excluding them for the elastic response option (NDERV=1) decreases the amount of memory required by $23{,}000_8$ cells.

The loader segmentation directives for DEPROP and DEPROB are listed in Tables 4 and 5 respectively. Note that the directives in Table 4 are applicable to an elastic-plastic panel solution since they contain SIGMA, SIGMAB, and CBLANK.

The Fortran (FTN) compiler has been used with the NOS/BE Version 1.2 Operating System to compile the codes on the CYBER 176 computer. Common blocks are given in Table 6. In order to reduce the LCM required and expedite running time, the user should always set the P and Q arrays to the amount of storage needed for the data tape, Load Option 4. During operation, pressure data that cannot be stored in P are automatically stored in Q. Check the MAXD and MAXD2 dimensions in subroutines PINIT and PRESS. Be sure there is enough storage capability for the number of pressures selected, based on the shock arrival time to the first gage, the TSTOP used, and the time interval resulting from the SKIP selected.

The program can currently handle 11914 pressures for each of 22 channels. For a sampling interval of 10 μs, 119 ms of pressure-time history can be stored in the program.

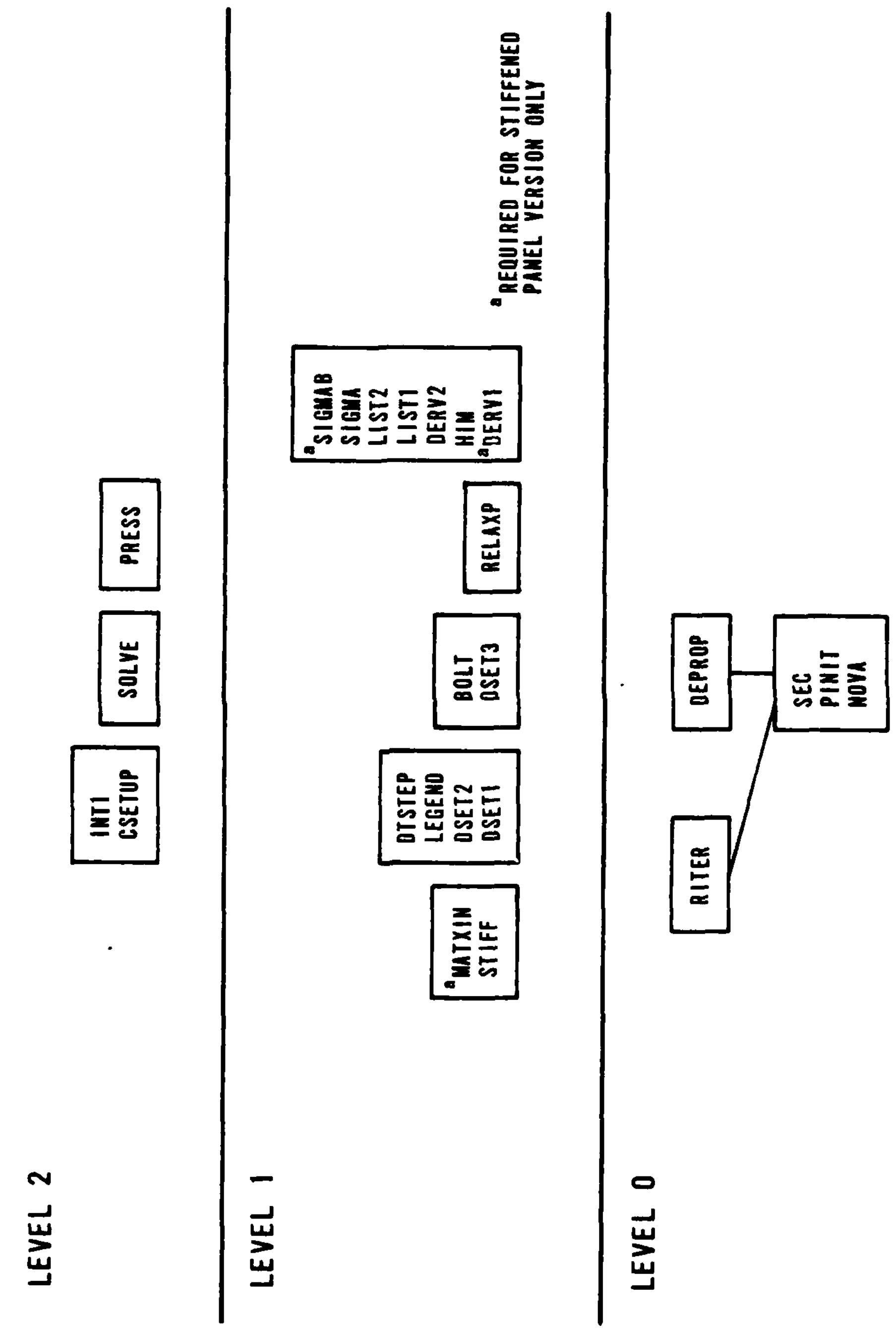

Figure 3. DEPROP Segmentation Tree Structure

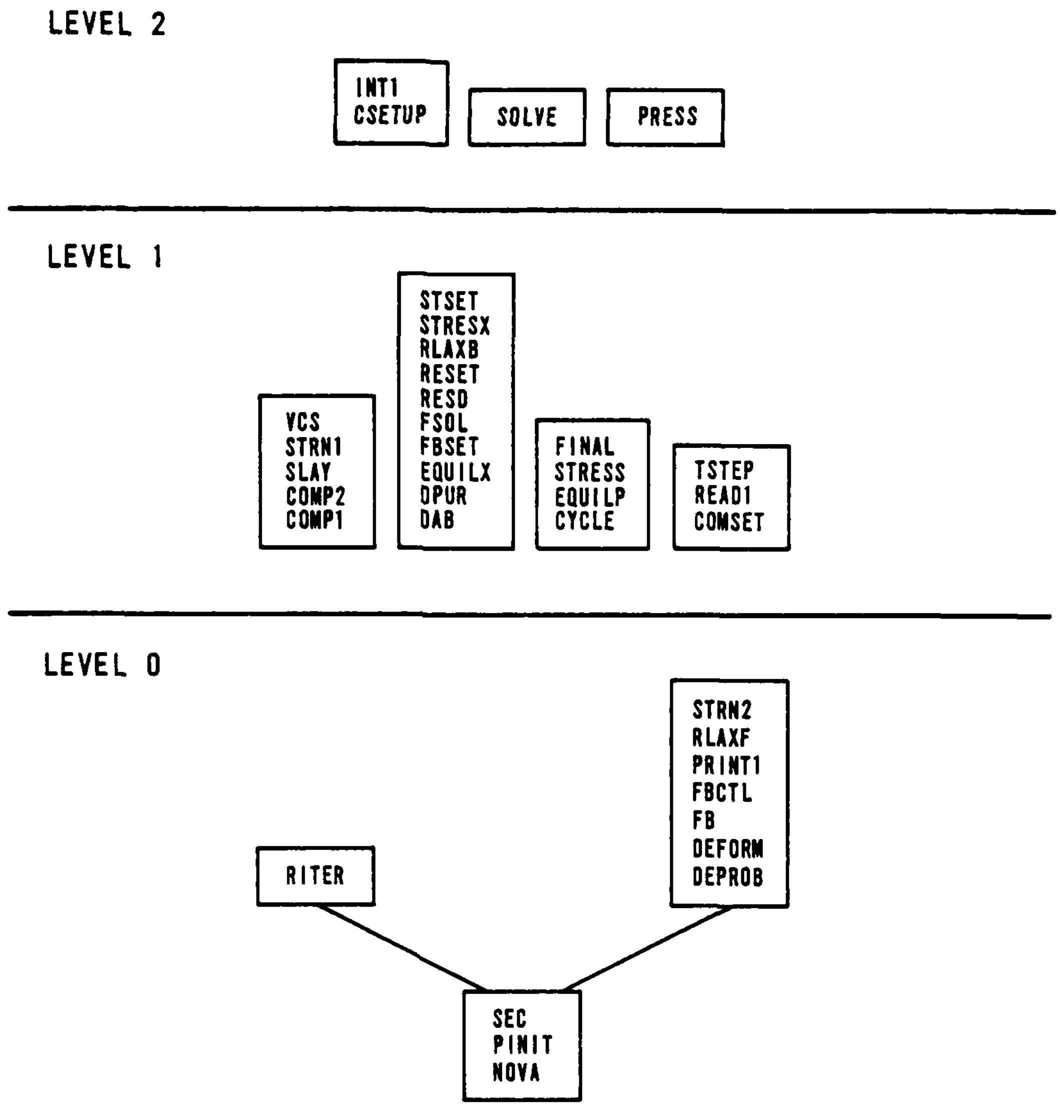

Figure 4. DEPROB Segmentation Tree Structure.

TABLE 3. COMMON BLOCKS GLOBAL AND SAVE IN SEGMENTATION

Common block ownership
for segmentation

Routine	Common Blocks	
[a]NOVA	CNOVA	CLOAD
	CBLK1	COM1
	COM2	
DEPROB	BLK2	
	BLK3	
	BLK6	
DEPROP	CBLK2	CBLK10
	CBLK3	CBLK11
	CBLK4	CBLK13
	CBLK5	[b]CBLK15
	CBLK7	[b]CBLK17
	[b]CBLK8	CBLANK
	CBLK9	
[b]DERV1	CBLK6	[b]CBLK16
RELAXP	CBLK12	

[a]The SAVE designation is not necessary for common blocks in the root segment.
[b]Required for stiffened panel version (NOVA-2S) only.

TABLE 4. DEPROP SEGMENTATION LOADER DIRECTIVES

	TREE	NOVA
NOVA	INCLUDE	PINIT, SEC
	LEVEL	
	TREE	RITER
	TREE	DEPROP
	LEVEL	
	TREE	DSETI
DSETI	INCLUDE	DSET2, LEGEND, DISTEP
	TREE	DSET3
DSET3	INCLUDE	BOLT
	[a]TREE	[a]STIFF
[a]STIFF	INCLUDE	[a]MATXIN
	TREE	RELAXP
	TREE	[a]DERV1
[a]DERV1	INCLUDE	HIM, DERV2, LIST1, LIST2, SIGMA, [a]SIGMAB
	LEVEL	
	TREE	CSETUP
CSETUP	INCLUDE	INT1
	TREE	PRESS
	TREE	SOLVE
NOVA	GLOBAL	CNOVA, CLOAD, CBLK1, COM1, COM2
DEPROP	GLOBAL	CBLK2, CBLK3, CBLK4, CBLK5, CBLK7, [a]CBLK8, CBLK9, CBLK10, CBLK11, CBLK13, [a]CBLK15, [a]CBLK17, CBLANK-SAVE
RELAXP	GLOBAL	CBLK12-SAVE
[a]DERV1	GLOBAL	CBLK6, [a]CBLK16-SAVE
	END	

[a]Required for stiffened panel version (NOVA-2S) only; otherwise replace DERVI with HIM and relocate CBLK6 to DEPROP GLOBAL declaration.

TABLE 5. DEPROB SEGMENTATION LOADER DIRECTIVES

	TREE	NOVA
NOVA	INCLUDE	PINIT, SEC
	LEVEL	
	TREE	RITER
	TREE	DEPROB
DEPROB	INCLUDE	DEFORM, FB, FBCTL, PRINT1, RLAXF, STRN2
	LEVEL	
	TREE	COMP1
COMP1	INCLUDE	COMP2, SLAY, STRN1, VCS
	TREE	DAB
DAB	INCLUDE	DPUR, EQUILX, FBSET, FSOL, RESD, RESET, RELAXB, STRESX, STSET
	TREE	CYCLE
CYCLE	INCLUDE	EQUIP, STRESS, FINAL
	TREE	COMSET
COMSET	INCLUDE	READ1, TSTEP
	LEVEL	
	TREE	CSETUP
CSETUP	INCLUDE	INTI
	TREE	SOLVE
	TREE	PRESS
NOVA	GLOBAL	CNOVA, CLOAD, CBLK1, COM1, COM2
DEPROB	GLOBAL	BLK2, BLK3, BLK6-SAVE
	END	NOVA

TABLE 6. LEVEL 2 VARIABLES FOR LARGE CORE MEMORY (LCM)

COMMON BLOCKS	ROUTINES	VARIABLES	DIMENSION-LENGTH	FTN LIMIT	LCM LIMIT
COM1	PINIT, PRESS	P(NGAGE,MAXD2)	22 x 5957 = 131,054	131,071	
COM2	PINIT, PRESS	Q(NGAGE, MAXD-MAXD2)	22 x 5957 = 131,054	131,071	360,000
CBLANK	SIGMA	ALTT(1156). . . TTNU(1156)	1156 x 14 = 16,184	131,071	

NOTES: 1. All dimensions in table are in decimal.

2. Common block variables in CBLANK can be located in small core memory (SCM) on the Control Data Corp 6600 computer with a net increase in core of 23,000 cells. Subroutine SIGMA is not required for beams or elastic panel response.

3. Common block variable P and Q are required only for load option 4; otherwise, they can be dimensioned at (1,1) in either SCM or LCM.

4. NGAGE represents the number of experimental channels of pressure data on the tape. MAXD is the total number of pressures to be stored in memory. MAXD2 is the number to be stored in P; the rest spill over into Q. MAXD can be calculated as follows: MAXD = 2 + (TSTOP-T1M1)/(SKIP • tape time interval).

5. For load option 4, if MAXD2 or MAXD is changed, then the two statements beginning at statement number 6000 in PINIT need to be changed.

6. Current limits on tape data are as follows:

   a. NGAGE - 22

   b. MAXD2 - 5957

   c. MAXD - 11914

Typical compiler execution times (using optimization for fast execution of object code, OPT=2) and the SCM and LCM required to load the codes are given in Table 7.

TABLE 7. MEMORY REQUIREMENTS, LOAD OPTION 4

CODE	[a]COMPILE TIME (S)	SCM (octal words)	[b]LCM (octal words)
[c]DEPROP	8.383	132,167	1,236,310
DEPROB	8.335	111,762	1,016,020

[a]OPT = 2, optimization for fast execution of object code.

[b]Contains CBLANK and COM1 and COM2 common blocks with P and Q set at current limits (22 x 5957).

[c]For elastic analyses only (SIGMA and SIGMAB routines omitted), SCM reduces to 130,513 words and LCM to 1,132, 360 words.

Once the code has been executed, an absolute object code file is created by the segmentation loader. Saving this file will eliminate the need to recreate an absolute file each time the code is used, unless changes are made to either the segmentation directives or the code. It is not necessary to preset the core to zero prior to execution.

Table 8 contains the Job Control Language (JCL) required to run NOVA-2LT (NOVA-2LTS) on the CYBER 176 computer, using Load Option 4.

5. INPUT DATA INSTRUCTIONS

Most of the required input data pertain to DEPROB or DEPROP, and are fully described in References 1 and 6. The general NOVA input data have been altered and are listed in Table 9.

All the data are organized in groups, with each group beginning on a separate data card. Additional cards may be required for a particular group. The format corresponding to each group is given in parentheses and is always in fields of 12. Each parameter in the data set is described and its units specified; the associated variable name is also given. Columns 73 and 80 can be used to label the cards in the data sets to facilitate assembly of the card deck and recognition of the variables in the data sets.

To accommodate the changes in DEPROP for the NOVA2S version (Ref. 6), the following modifications were made in the DEPROP input data.

In Group 5, page 114 of Reference 6, the variable NBND (boundary condition code) is defined as

NBND = PQ

where P and Q indicate the numerical codes for the boundary condition in the γ and β directions as follows:

Code.	Edge Condition
1	clamped-clamped
2	simple-simple
3	free-free
4	clamped-simple
5	clamped-free

Values of PQ may range from 11 to 55.

In Group 17, page 18 of Reference 6, new groups A and B are added as follows:

Group 17A: (I12) NSPR

Number of linear elastic springs. (NSPR)

If NSPR = 0, skip Group 17B.

Group 17B: (3I12, F12.1) IDIR(I), NSPG(I), NSPB(I), BIGK(I)

Code designating direction of spring (IDIR):

1 u-direction
2 v-direction
3 w-direction

Number indicating the gamma (γ_j) spring position. (NSPG)

Number indicating the beta (β_k) spring position. (NSPB)

Linear elastic spring stiffness (K, lb/in). (BIGK)

Repeat Group 17B for I - 1, NSPR. The cards in Group 17B may be arranged in any order.

Group 23 establishes a correspondence between the location of the pressure data channels on the data tape and the mass points of the analytical model. The numbers which specify the location of the gages relative to the mass points are ordered from the first mass point for beam models and from the coordinate origin for panel models. As an example, Figure 5 shows a simple beam model

TABLE 8. JOB CONTROL LANGUAGE

Job Control Sequence	Comments
Job Card	
Account Card	
ATTACH, OLDPL,••••	UPDATE old program library file
UPDATE, W, Q.	Sequential library, quick compile
RETURN, OLDPL.	
FTN, A, I, LCM=I, OPT=2, PL=XXX, • • •	If LCM exceeds 131,071 words, LCM=I is required. If line print limit exceeds default, LP=XXX is required.
RETURN, COMPILE.	
REQUEST, ABS,*PF.	Ensures proper file residence for permanent file ABS
SEGLOAD, B=ABS.	
LDSET, PRESET=ZERO, MAP=BSEX,• • •	
LOAD, LGO.	
NGO.	
REWIND, ABS.	
REQUEST, TAPE10, • • • •	Pressure Data Tape
ABS.	Execute Code
CATALOG, ABS, • • • •	Absolute object code permanent file for future jobs
EXIT, S.	
EXIT.	
7/8/9	
UPDATE Directives	
7/8/9	
Segmentation Directives	
7/8/9	
Run Data	
6/7/8/9	

TABLE 9. NOVA-2LT (NOVA-2LTS) INPUT DATA

Group 1: (I12) NCASES

Number of cases to be run.

Group 2: (20A4) TITLE

Identifying title

Group 3: (5I12) ICOMP, KTYPE, KDAM, KDS, NDBUG

Structural Element Position Code (ICOMP)

1 Nonfuselage element, or at least no additonal skin effect.

2 Fuselage element. A stringer, longeron or frame will derive additional skin support against internal pressurization. (See Vol. I of Ref. 1, section 4.1.8, for additional discussion).

Structural Element Code (KTYPE)

1 Single layer metal panel
2 Single Layer plastic panel
3 Honeycomb metal panel
4 Honeycomb plastic panel
5 Multilayer plastic panel
6 Stringer or longeron
7 Frame
8 Metal ring
9 Plastic ring (radome)
10 Metal rib

Damage Criteria Code (KDAM)

0 No permanent damage (threshold)
1 Catastrophic damage
2 No criteria, response run only
100 Noniteration, but program accumulates maximum values for threshold damage criteria.
101 Noniteration, but program accumulates maximum values for catastrophic damage criteria.

(Note--KDAM must be 0, 100, or 101 if NLOAD = 4 in Group 6).

Response option code (KDS)

1 Static only
2 Dynamic only
3 Static and dynamic

(Note--Only the dynamic option is available for KTYPE=10).

TABLE 9. CONTINUED.

Debug option (NDBUG)

0 No debug output (default option)
1 Most debug output
2 All debug output

If KDAM=2 or KDS=1, skip Group 4

Group 4: (F12.1) PDAM

Probability of exceeding specified damage level, expressed as a fraction. (0 <PDAM < 1.0)

For KTYPE < 6, use DEPROP input here. (Groups 1-14, Ref. 1)

For KTYPE > 5, use DEPROB input here. (Groups 1-22, Ref. 1)

If KDS = 2, skip Group 5.

Group 5: (F12.1) PS

Uniform static preload pressure, psi.

If KDS = 1, skip Groups 6-20

Group 6: (I12) NLOAD

Dynamic load option (NLOAD)

1 Uniform spatially, linear-exponential temporal function.
2 Uniform spatially, point-by point temporal description.
3 General, discrete load description (Not appropriate for KTYPE = 10).
4 Data tape. Evenly spaced times, one-dimensional variation spatially.

If NLOAD = 2, skip to Group 8.

If NLOAD = 3, skip to Group 10 or 15, depending on KTYPE

If NLOAD = 4, skip to Group 21

Group 7: (6F12.1) PP1, PPO, TTO, TPRIME, AA, ANN

Pressure, P_1, psi (PP1)

Pressure, P_o, psi (PPO)

Time, t_o, sec (TTO)

Time, t', sec (TPRIME)

TABLE 9. CONTINUED.

Constant, a, dimensionless (AA)

Constant, n, dimensionless (ANN)

(Note--See Figure 1 for definition of parameters).

Skip Groups 8-23

Group 8: (I12) NTIME

Number of points to be specified in point by point load description. Be sure to include time = 0. If time exceeds last value in table, the the last value of pressure is used. Dimensioned for 20 times and pressures.

Group 9: (2F12.1) TT(I), PT(I)

Time table, sec	(TT(I))
Pressure table, psi	(PT(I))

Repeat Group 9 for I = 1, NTIME

Skip Groups 10-23

If KTYPE < 6, skip to Group 15

Group 10: (112) NPS

Number of pressure stations for which there are valid data. Must have at least 2 stations, but not more than 41.

Group 11: (6F12.1) SP(I), I = 1, NPS

Position of pressure station relative to mass point locations in DEPROB. For example, a station 2/3 of the distance between mass numbers 3 and 4 in the DEPROB model would mean

$$SP = 3.67. \quad (0 \leq SP \leq N + 1).$$

(Note--Do not assign an SP(I) to bad channels).

Group 12: (6112) LTIME(I), I = 1, NPS

Number of entries in pressure versus time table for each spatial location. LTIME must be at least 2, but not more than 14.

Group 13: (6F12.1) TTB(K,I), K = 1, LTIME(L)

Break point times for each spatial station, sec.

TABLE 9. CONTINUED.

Group 14: (6F12.1) PRTB (K, I, LTIME(I)

Break point overpressures corresponding to times in Group 13 for each spatial station, psi.

Repeat Groups 13-14 for each station, I = 1, NPS

Skip Groups 15-23

Group 15: (2I12) NPX, NPY

Number of pressure stations in gamma direction. Must have at least 2 stations, but not more than 10. (NPX)

Number of pressure stations in beta direction. Must have at least 2 stations, but not more than 10. (NPY)

(Note--Since the pressure mesh is rectangular, NPX·NPY pressure records must be supplied).

Group 16: (6F12.1) XP(I), I = 1, NPX

x-positions at which time histories are specified, in. (XP)

Group 17: (6F12.1) YP(I), I = 1, NPY

y-positions at which time histories are specified, in or deg. (YP)

Group 18: (6F12.1) KTIME (J, I), J=1, NPY

Number of entries in pressure versus time table for each spatial point.

REPEAT Group 18 for I = 1, NPX

Group 19: (6F12.1) TTP (K, J, I), K=1, KTIME (J,I)

Break point times for each spatial station, sec.

Group 20: (6F12.1) PRT(K,J,I), K=1, KTIME(J,I)

Break point overpressure corresponding to times in Group 19 for each spatial station, psi.

REPEAT Groups 19 and 20 for each station, with the y-index varying more rapidly: J=1, NPY; then I=1, NPX.

Skip Groups 21-23

Group 21: (2F12.1) TIM1, SKIP

Start time, relative to the data tape time scale, sec. Time should be as near as possible to the first shock engulfment.

TABLE 9. CONTINUED.

Skip frequency, dimensionless. The time history will be sampled at this frequency. A 1.0 will mean every time is used; 3.0 means every 3rd time, etc. Must be a whole number.

Group 22: (I12) NGAGE

Number of channels of data on the tape. Each channel represents a pressure gage and a tape file.

Group 23: (6I12) NORDER(I), I=1, NGAGE

Gage location code. The numbers which specify the location of the gages relative to the mass points in the model are ordered from beginning of DEPROB model (mass point 1) or coordinate origin in DEPROP. A zero indicates data for a specific channel is not to be used. The example in Figure 5 defines the code for a beam where the data tape was generated backward with four gages. Therefore, the pressure data from gage number 4 is the first data channel written on the tape, and is designated NORDER(1). Note that gage number 2 was omitted since it was found to be faulty. Table 10 is constructed in the same order as the tape was written; the Ith index of NORDER corresponds to the Jth index of SP.

If KTYPE < 5, the next input is identical to that described in Groups 15, 16, and 17. The only limitation is that either NPX or NPY (or both) must be equal to 1. If they are both 1, for a uniform distribution, then Groups 16 and 17 can be two blank cards.

If KTYPE > 5, the next input is identical to that described in Groups 10 and 11. The only difference is that NPS can be 1 for a uniform distribution. In that case Group 11 can be a blank card.

REPEAT Groups 2-23 for each additional case, as specified in Group 1. If a data tape was used (NLOAD=4), the tape will be rewound and used with the next case for which NLOAD=4 is specified.

with pressure gage locations and Table 10 lists the pressure gage order and locations relative to the mass points. In this case the data type was generated backwards using test data from four gages.

6. EXPERIMENTAL PRESSURE DATA (LOGICAL FILE TAPE10).

The following procedures are suggested for generating an experimental pressure data tape (Logical File TAPE10) for use with NOVA-2LT (NOVA-2LTS):

a. Determine, as accurately as possible, the shock arrival time for the first gage to be engulfed.

b. Digitize data for all gages approximately 10 μs prior to and 140 μs following the shock arrival time determined above.

c. Digitize all data at 10 μs intervals.

d. If possible, order data on tape for all gages used, assuming this is consistent with the DEPROB model. Not imperative, however.

e. Output data tape should have time in microseconds and overpressure in psi. Tape should be unlabelled, unformatted, stranger(S), at 800 bpi using Block Type K and Record Type U.

f. Check pressure plots for any suspect type gages. If data is bad the gage can be eliminated during execution of the NOVA-2LT (NOVA-2LTS) code by means of card input data.

g. Determine Δt for NOVA-2LT. Then select a sampling frequency (SKIP) for the data tape. For example, if Δ = 30 μs and the tape is digitized every 10 μs, a SKIP parameter of at least 3.0 should be used. Larger values should be tried for one case to determine the largest appropriate SKIP.

The format for writing the digitized data on tape is given in Table 11.

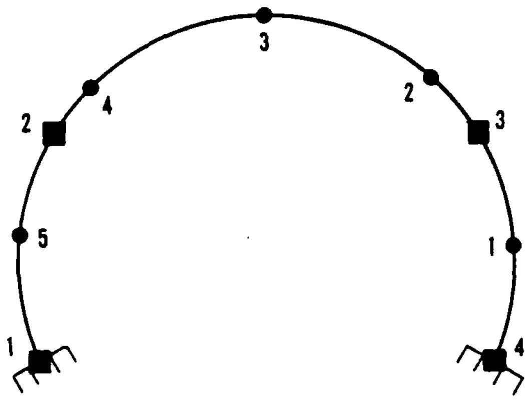

Figure 5. DEPROB analytical model with pressure gage locations.

TABLE 10. PRESSURE GAGE ORDER AND LOCATION

Gage No. I=1,NGAGE	Gage Order NORDER (I)	Gage Location [a]J	[b]SP(J)
1	3	1	0.0
2	0		
3	2	2	1.5
4	1	3	6.0

[a]The J indices (with the corresponding SP's) must be ordered in increasing value.

[b]Number of gage locations (SP's) must equal the number of nonzero values in the NORDER(I) column.

TABLE 11. DATA TAPE FORMAT

Record	No. of Words	Description of Parameter
EOF	1	
Identification	100	Mixture of both fixed and real numbers. Need to equivalence both types of variables, e. g., ID(100), DD(100).
DATA for Channel 1	ID(17)*ID(18) (usually 2*252)	252 data pairs consisting of time (μs) and pressure (psi), one after another. Real-time numbers.
Last Data Record - Channel 1	Same as other data records	Same. Last record may be padded with OCTAL constant 3 followed by nineteen 4's.
EOF	1	
Data for Channel 2	Same as other data records	Same
Last Data for Channel NGAGE	Same as other data records	Same
EOF	1	
TRAILER Record	10	Each word is made up of 10 F's
EOF	1	
EOF	1	

IV. GEOMETRIC APPLICATIONS OF RADIAL IMPERFECTIONS

The DEPROP modal solution in the NOVA-2S computer code assumes that the analyzed panel has a partial right-cylindrical geometry. For many aircraft structural applications, the panels of interest are smooth but arbitrarily curved. The analysis of a smoothly curved, noncylindrical panel can be attempted with a circular approximation for which the radial distance from the circle to the actual curve is minimized. The use of this circular approximation may be adequate for analyses in which large deformations are encountered (e. g., catastrophic failure analysis), but may prove to be misleading in elastic or threshold-of-permanent damage analyses in which the initial geometry plays a more significant role in the ensuing deformation profile and time history. For these situations a more accurate model of the panel curvature can be derived by placing initial radial imperfections on the approximated circle to conform with the actual geometry of the panel.

These initial radial imperfections are applied by specifying initial preload amplitudes for the DEPROP modes used to approximate the radial deflection (w-direction) of the analyzed panel. The proper summation of these initial model amplitudes will allow the original circular model to represent the actual panel curvature more accurately.

As a first approximation the user may be able to determine the magnitude of the modal coefficients needed by superimposing the circular approximation on the actual curve and measuring the amplitude of the major modal imperfections. For example, in Figure 6, the value δ is the amplitude of the second symmetric mode which is the apparent dominant mode of the actual shape. Because of the sign convention in DEPROP, the amplitude δ would be assigned a negative value (for the second symmetric mode) in the input data; i.e., FG(N,M), Group 13 in NOVA-2LT and Group 30 in NOVA-2LTS.

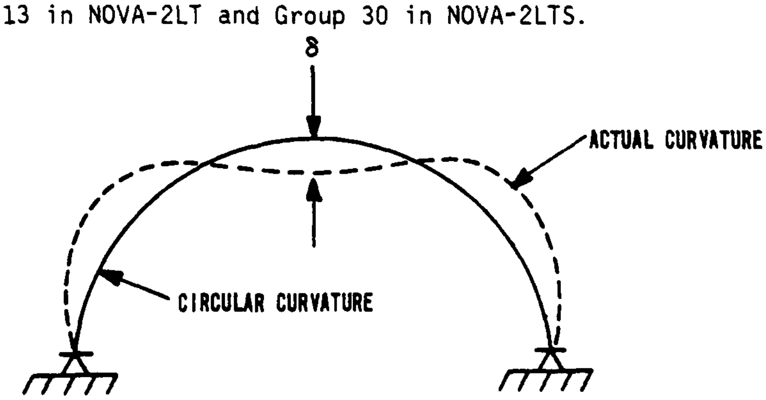

Figure 6. Panel curvature, idealized vs actual

The individual modal amplitudes may be calculated more accurately by the utilization of a Fourier analysis as presented in Reference 8. The Fourier method applies a least-squares fit to both the actual and the approximating periodic functions. For the DEPROP analysis, the approximating periodic functions are the assumed modes defined over the interval (0 to π) as given in Reference 1.

First, the approximating circle must be fitted to the actual panel curvature. This may be done by solving the following simultaneous equation for Y_o, Z_o, and R in the Y-Z rectilinear coordinate system.

$$\begin{aligned}(Y_o - Y_1)^2 + (Z_o - Z_1)^2 &= R^2\\ (Y_o - Y_2)^2 + (Z_o - Z_2)^2 &= R^2\\ (Y_o - Y_3)^2 + (Z_o - Z_3)^2 &= R^2\end{aligned} \qquad (5)$$

where

Y_o, Z_o	Coordinates of the circle's origin.
Y_1, Z_1	Coordinates of the two end points of the actual curve.
Y_2, Z_2	
Y_3, Z_3	User determined coordinates of a third point to be located on the approximating circle.
R	Radius of approximating circle.

The approximating circle is now used to determine the amplitudes of the initial modal imperfections through the use of a Fourier analysis. A function $f(\beta)$ is defined as the radial distance from the circle to the actual curve (radially inward distances are positive). This function is then approximated by a summation of periodic functions, $G_k(\beta)$, and their individual amplitudes, A_k.

$$f(\beta) = \sum_{k=1}^{N} A_k \, G_k \, (\beta) \qquad (6)$$

The values of A_k are determined by applying a least-squares fit over the interval 0 to π (the interval over which the DEPROP modes are defined) in the following manner:

$$\int_0^{\pi} \left[f(\beta) - \sum_{k=1}^{N} A_k\, G_k(\beta)\right]^2 \quad d\beta = \text{minimum} \tag{7}$$

from which the following condition must be satisfied:

$$\int_0^{\pi} \left(f(\beta) - \sum_{k=1}^{N} A_k\, G_k(\beta)\right) \left(\sum_{j=1}^{N} G_j(\beta)\right) d\beta = 0 \tag{8}$$

When the orthogonality constraint

$$\int_0^{\pi} G_j(\beta)\, G_k(\beta)\, d\beta = 0 \text{ for } j \neq k$$

is introduced into Equation 8, the following relation is obtained

$$\int_0^{\pi} G_k(\beta)\, f(\beta)\, d\beta = A_k \int_0^{\pi} \left(G_k(\beta)\right)^2 \quad d\beta \tag{9}$$

However, the function $f(\beta)$ is not easily defined over the continuous domain, requiring that a discrete domain approximation be applied by defining $f(\beta_r)$. The function $f(\beta_r)$ is a set of values of $f(\beta)$ tabulated at equal intervals β_r. The range of r is from 1 to P for which P-1 independent values of $f(\beta)$ are determined. For the end points it is assumed that $f(1) = f(P) = 0$.

Equation 10 then takes the form

$$\sum_{r=1}^{P-1} G_k\left(\beta_r\right) f\left(\beta_r\right) = A_k \sum_{r=1}^{P-1} \left(G_k(\beta_r)\right)^2 \tag{10}$$

and each value of A_k is given by

$$A_k = \frac{\sum_{r=1}^{P-1} G_k\left(\beta_r\right) f\left(\beta_r\right)}{\sum_{r=1}^{P-1} \left(G_k\left(\beta_r\right)\right)^2} \qquad \text{(For } k = 1 \text{ to } N\text{)}$$

where N is the number of modes used in the approximation.

The orthogonal modes are defined below for each set of boundary conditions in DEPROP.

a. Simply supported at both ends:

$G_k(\beta_r) = \sin(k\beta_r)$

b. Clamped-clamped

$$G_k(\beta_r) = \cosh\left(\frac{\lambda_k\beta_r}{\pi}\right) - \cos\left(\frac{\lambda_k\beta_r}{\pi}\right) - \left[\alpha_k \sinh\left(\frac{\lambda_k\beta_r}{\pi}\right) - \sin\left(\frac{\lambda_k\beta_r}{\pi}\right)\right] \quad (11)$$

where

λ_k : roots of $\cos \lambda_k \cosh \lambda_k = 1$

$$\alpha_k = \frac{\cosh \lambda_k - \cos \lambda_k}{\sinh \lambda_k - \sin \lambda_k}$$

Using the following trigonometric identities,

$$\sinh z = \frac{e^z - e^{-z}}{2} \qquad \cosh z = \frac{e^z + e^{-z}}{2}$$

and defining $\bar{\lambda}_k = \dfrac{\lambda_k}{\pi}$

$$G_k(\beta_r) = \left(\frac{1-\alpha_k}{2}\right) e^{\bar{\lambda}_k\beta_r} \left(\frac{1 + \alpha_k}{2}\right) e^{-\bar{\lambda}_k\beta_r} - \cos \bar{\lambda}_k\beta_r + \alpha_k \sin \bar{\lambda}_k\beta_r \quad (12)$$

c. Clamped-simply supported

The same as clamped-clamped except that the definition of λ_k is as follows:

λ_k : roots of $\tan \lambda_k = \tanh \lambda_k$

The modes are all independent of each other so that the calculation of A_k for one value of k is independent of any other value of k. Because of the characteristics of the least-squares fit and to the numerical accuracies of the iterated values for λ_k and α_k, a fine mesh of points must be chosen for β_r. An amplitude should be calculated for each mode used in the analysis, and the values of A_k back substituted into the initial equation.

$$f(\beta_r) = \sum_{k=1}^{N} A_k \; G_k(\beta_r) \quad (13)$$

to determine the loss of accuracy due to numerical round offs.

Note that the calculated amplitudes of the initial imperfections, A_k, referred only to the modes in the β direction. The calculated values of A_k are therefore associated with the coefficients of the <u>first</u> λ mode and the

kth β mode. This will give an accurate representation of the actual panel curvature only at the center of the assumed model, and the accuracy will decrease as the λ boundaries are approached However, the analysis can be applied to the entire surface of the panel and include both the λ and β imperfection modes. The λ modes would be used to preserve an approximate straight shape in the λ direction over the noncircular cross-section, except at the λ boundaries where by definition the imperfections are zero. The initial approximation, presented in Equation 6, is more generally given by the following equation:

$$f\left(\lambda_s,\beta_r\right) = \sum_{k=1}^{N} \sum_{\ell=1}^{M} A_{\ell k} G_k\left(\beta_r\right) H_\ell\left(\lambda_s\right) \tag{14}$$

By applying the analogous solution procedure given in Equations 7-10, the value of the initial imperfection modal coefficients, $A_{\ell k}$, are given in Equation 15.

$$\sum_{r=1}^{P-1} \sum_{s=1}^{Q-1} G_k\left(\beta_r\right) H_\ell\left(\lambda_s\right) f\left(\lambda_s,\beta_r\right) = A_{\ell k} \sum_{r=1}^{P-1} \sum_{s=1}^{Q-1} \left(G_k\left(\beta_r\right) H_\ell\left(\lambda_s\right)\right)^2 \tag{15}$$

for which $H_\ell\left(\lambda_s\right)$ is the ℓth mode in the λ direction, and it is analogous to the β modes described above for the applicable boundary conditions.

The λ function is divided into equal intervals such that $s = 1$ to Q and $A_{\ell k}$ is the coefficient of the ℓth λ mode and the kth β mode.

REFERENCES

1. Mente, L. J. and Lee, W. N., NOVA-2 A Digital Computer Program for Analyzing Overpressure Effects on Aircraft, Parts 1 and 2, AFWL-TR-75-262, Air Force Weapons Laboratory, Kirtland Air Force Base, NM., August 1976.

2. Lee, W. N., The Structural Response Codes of NOVA-2 as Modified to Include User-Designated Pressure Functions, KA-TM-103, Kaman AviDyne, Burlington, MA., June 1976.

3. Lee, W. N., A Users' Manual for NOVA-2L, KA-TM-109, Kaman AviDyne, Burlington, MA., May 1977.

4. Syring, R. F. and Pierson, W. D., Structural Response to Simulated Nuclear Overpressures (STRESNO): A Test Program Establishing A Data Base for Evaluating Present and Future Analytical Techniques DNA 4278F-1, Defense Nuclear Agency, March 1977.

5. Lee, W. N., A Users' Manual for NOVA-2LT, KA-TM-114, Kaman AviDyne, Burlington, MA., January 1978.

6 Mente, L. J. and Lee, W. N., NOVA-2S, A Stiffened Panel Extension of the NOVA-2 Computer Program, AFWL-TR-78-182, Air Force Weapons Laboratory, Kirtland Air Force Base, NM., December 1978.

7. Stagliano, T. R., Mente, L. J., Lee, W. N., Memorandum on the Support Work for the STRESNO Phase II Test Program, KA-TM-118, Kaman Avi Dyne, Burlington, MA., March 1979.

8. Hildebrand, F. B., Introduction to Numerical Analysis, 2nd Ed., McGraw Hill, New York, N. Y. 1974.

APPENDIX A

DYNAMIC LOAD OPTION CHANGES

This appendix contains the changes necessary to convert the NOVA-2 code to the NOVA-2LT version and replaces the nuclear blast loading model with four transient pressure models and one static pressure model.

Since the NOVA-2 code library is maintained at the AFWL using the Control Data Corporation 6000/7000 UPDATE program, the UPDATE directives needed to accomplish these changes are also included.

CHANGES TO CONVERT
NOVA-2 TO NOVA-2LT TAPE VERSION

```
*IDENT JAN0178

*DELETE LIST1.158
     1  11X,5HV(IN),11X,5HW(IN),7X,14HPRESSURE (PSI))
*DELETE LIST2.165
     1  11X,5HV(IN),11X,5HW(IN),7X,14HPRESSURE (PSI))
*INSERT SIGMA.17
      LEVEL 2, ALTT
*PURDECK IODUM
*PURGE IODUM
*PURDECK CLOAD
*PURGE CLOAD
*ADDFILE INPUT,CNOVA
*COMDECK CLOAD
      COMMON /CLOAD/ PP1,PFC,TTC,TPRIME,AA,ANN,OTT1,OTTO,AZ,
     1  JL,NTIME,NLOAD,PT(20),TT(20), KTIME(10,10) LTIME(41),ISP(40),
     2 JLB(41),NPS,TTP(6,10,10),PRT(6,10,10),NPX,NPY,XP(22),YP(22),
     3  IXI(23),JYJ(23),JLT(10,10),PRTT(10,10),DX1(23),DY1(23),
     4  NGSUM,DEL,MAXD2,MAXC,PS
*COMDECK COM1
      COMMON /COM1/ P(22,5957)
      LEVEL 2, P
*COMDECK COM2
      COMMON /COM2/ Q(22,5957)
      LEVEL 2, Q
*PURDECK NOVA
*PURGE NOVA
*ADDFILE INPUT,CBLANK
*DECK NOVA
      PROGRAM NOVA  (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1=513,
     1  TAPE10=1)
C
C     THIS IS THE NOVA-2LT VERSION OF NOVA.  THE AERODYNAMIC AND BLAST
C     ROUTINES ARE REPLACED BY USER DESIGNATED PRESSURE FUNCTIONS.
C     PROVISION HAS ALSO BEEN MADE FOR READING PRESSURE DATA FROM TAPE.
C     NOVEMBER,1977.
C
*CALL CLOAD
*CALL CNOVA
C
    1 FORMAT(6I12)
    2 FORMAT(6F12.1)
    3 FORMAT (20A4)
      NCASE = 0
      INOUT = 1
      READ(5,1) NCASES
  100 READ(5,3) (TITLE(I),I=1,20)
      NCASE = NCASE + 1
      KERR = 0
      NTRIAL = 0
      READ (5,1) ICOMP,KTYPE,KDAM,KDS,NDBUG
      IF (KDS.EQ.1) KDAM = 2
```

```
      NCHPT = 0
      IF (KDAM.LT.2) NCHPT = 1
      IF (KDAM.GT.2) KDAM = KDAM - 100
      IF(KDAM.LT.2) READ (5,2) PDAM
      IF(INOUT.EQ.0) GO TO 1400
      WRITE(6,3000) (TITLE(I),I=1,20)
      IF (KTYPE.LT.6.OR.KTYPE.GT.7) ICOMP = 5
      IF (ICOMP.EQ.2) WRITE (6,5000)
      IF (NCHPT.EQ.1) WRITE (6,3100)
      IF (NCHPT.EQ.0) WRITE (6,3200)
      IF(KDAM.EQ.0) WRITE(6,3300)PDAM
      IF(KDAM.EQ.1) WRITE(6,3400)PDAM
      GO TO (300,400,500,600,700,800,900,1010,1000,1020),KTYPE
  300 WRITE(6,3500)
      GO TO 1050
  400 WRITE(6,3600)
      GO TO 1050
  500 WRITE(6,3700)
      GO TO 1050
  600 WRITE(6,3800)
      GO TO 1050
  700 WRITE(6,3900)
      GO TO 1050
  800 WRITE(6,4000)
      GO TO 1050
  900 WRITE(6,4100)
      GO TO 1050
 1000 WRITE(6,4200)
      GO TO 1050
 1010 WRITE(6,4700)
      GO TO 1050
 1020 WRITE(6,4800)
 1050 GO TO (1100,1200,1300), KDS
 1100 WRITE(6,4300)
      GO TO 1400
 1200 WRITE(6,4400)
      GO TO 1400
 1300 WRITE(6,4500)
 1400 NCALL = 2
      IF(KTYPE.GT.5) CALL CEPROB
      IF (KTYPE.LT.6) CALL CEPROP
      IF(KERR.GT.0) GO TO 1600
      NCALL = 1
      CALL PINIT(0)
      IF(KTYPE.GT.5) CALL CEPROB
      IF (KTYPE.LT.6) CALL CEPROP
      IF(KDS.EQ.1) GO TO 1600
      IF(KERR.GT.0) GO TO 1600
      NCALL = 0
      KCK = 0
```

```
      CALL PINIT(1)
      IF (KERR.GT.0) GO TO 1700
      RTRIAL(1)=1.0
 1500 NTRIAL = NTRIAL + 1
      IF(KDAM.LT.2) WRITE(6,4600) NCASE,NTRIAL,RTRIAL(1)
      IF (KTYPE.GT.5) CALL DEPROP
      IF (KTYPE.LT.6) CALL DEPROP
      IF(KERR.NE.0) GO TO 1600
      IF (NCHPT.EQ.0) GO TO 1600
      CALL RITER (CRIT,RTRIAL,NTRIAL,8,KOK)
      IF(KOK.EQ.0) GO TO 1500
 1600 IF (NCASE.LT.NCASES) GO TO 100
 1700 STOP
C
 3000 FORMAT (1H1,30X,15HN C V A - 2 L T//1X,20A4)
 3100 FORMAT (14H ITERATION RUN)
 3200 FORMAT (32H RESPONSE RUN ONLY, NO ITERATION)
 3300 FORMAT (42H NO DAMAGE LEVEL, PROBABILITY OF EXCEEDING, F6.3)
 3400 FORMAT (52H CATASTROPHIC DAMAGE LEVEL, PROBABILITY OF EXCEEDING,
     1F6.3)
 3500 FORMAT (28H0SINGLE-LAYER METAL PANEL  )
 3600 FORMAT (30H0SINGLE-LAYER PLASTIC PANEL  )
 3700 FORMAT (25H0HONEYCOMB METAL PANEL  )
 3800 FORMAT (27H0HONEYCOMB PLASTIC PANEL  )
 3900 FORMAT (29H0MULTI-LAYER PLASTIC PANEL  )
 4000 FORMAT (30H0METAL STRINGER OR LONGERON  )
 4100 FORMAT (15H0METAL FRAME  )
 4200 FORMAT (16H0PLASTIC RING  )
 4300 FORMAT (21H0STATIC SOLUTION ONLY)
 4400 FORMAT (22H0DYNAMIC RESPONSE ONLY)
 4500 FORMAT (37H0STATIC SOLUTION AND DYNAMIC RESPONSE)
 4600 FORMAT (12H1CASE NUMBER I2/
     114H TRIAL NUMBER I3, 10X, 18H  RANGE FACTOR = E14.6)
 4700 FORMAT(11H0METAL RING)
 4800 FORMAT(13H0RIB BUCKLING)
 5000 FORMAT (69H0STRUCTURAL ELEMENT DOES DERIVE ADDITIONAL SUPPORT FROM
     1 FUSELAGE SKIN)
      END
*PURDECK PINIT
*PURGE PINIT
*ADDFILE INPUT,CSETUP
*DECK PINIT
      SUBROUTINE PINIT(M)
*CALL CNOVA
*CALL CLOAD
*CALL CBLK1
*CALL COM1
*CALL COM2
      DIMENSION ID(100),DC(100),AI(1004),NORDER(22)
      EQUIVALENCE (ID(1),DC(1))
```

```
      DIMENSION TTB(14,41),PRTB(14,41),SP(41),SPS(41)
      EQUIVALENCE (PRT(1,1,1),PRTB(1,1)), (TTP(1,1,1),TTB(1,1))
      DATA TRD/10HFFFFFFFFFF/
C
      IF(M.EQ.1) GO TO 200
      PS=0.0
      IF(KDS.EQ.2) GO TO 150
C
C     STATIC
C
      READ(5,2000) PS
      WRITE(6,2200) PS
      NU=1
      PPP=PS
      IF (KTYPE.LT.6) GO TO 150
      IF (KTYPE.LT.10) GO TO 50
      DO 30 I=1,NMASS
   30 PB(I) = 0.
      GO TO 150
   50 DO 100 I=1,NMASS
  100 PB(I) = PS
  150 RETURN
C
C     DYNAMIC
C
  200 IF(KDS.EQ.1) GO TO 400
      READ (5,2050) NLOAD
      WRITE (6,2400) NLOAD
      GO TO (250,500,800,6000), NLOAD
  250 READ(5,2000) PP1,PPO,TTO,TPRIME,AA,ANN
      WRITE(6,2300) PP1,PPO,TTO,TPRIME,AA,ANN
      NU=1
      IF(TPRIME.EQ.0.0) GO TO 300
      PPRIIME=PPO*(1.0 - TPRIME/TTO)**ANN
      PPRIME = PPRIME*EXP(-AA*TPRIME/TTO)
      TT1=TPRIME*PP1/(PP1-PPRIME)
      OTT1=1.0/TT1
  300 OTTO=1.0/TTO
      AZ=AA*OTTO
  400 RETURN
C
  500 READ (5,2050) NTIME
      READ(5,2100) (TT(I),PT(I),I=1,NTIME)
      WRITE (6,2500) NTIME,(TT(I),PT(I),I=1,NTIME)
      JL = 2
      NU = 1
      RETURN
C
  800 IF (KTYPE.GT.5) GO TO 1000
C     PANELS.
C
```

```
      READ (5,2050) NPX,NPY
      WRITE (6,2700) NPX,NPY
      READ (5,2000) (XP(I),I=1,NPX)
      WRITE (6,3100) (XP(I),I=1,NPX)
      READ (5,2000) (YP(J),J=1,NPY)
      WRITE (6,3200) (YP(J),J=1,NPY)
      WRITE (6,3300)
      DO 820 I=1,NPX
      READ (5,2050) (KTIME(J,I),J=1,NPY)
  820 WRITE (6,2800) (KTIME(J,I)J=1,NPY)
      DO 840 I=1,NPX
      DO 840 J=1,NPY
      NTIME = KTIME(J,I)
      READ (5,2000)       (TTP(K,J,I),K=1,NTIME)
      WRITE (6,3600) I,J,(TTP(K,J,I),K=1,NTIME)
      WRITE (6,2900)
      READ (5,2000) (PRT(K,J,I),K=1,NTIME)
  840 WRITE (6,3000) (PRT(K,J,I),K=1,NTIME)
C     SPATIAL INTERPOLATION-EXTRAPOLATION.  INDICES ARE LOWER BOUND.
      DO 900 I=1,NGT
      DO 860 III = 1,NPX
      IF (XP(III).GT.XG(I)) GO TO 880
  860 CONTINUE
      III = NPX
  880 IF (III.GT.1) III = III - 1
      DX(I) = (XG(I) - XP(III))/(XP(III+1) - XP(III))
  900 IXI(I) = III
      DO 960 J = 1,NBT
      DO 920 JJJ = 1,NPY
      IF (YP(JJJ).GT.XB(J)) GO TO 940
  920 CONTINUE
      JJJ = NPY
  940 IF (JJJ.GT.1) JJJ = JJJ -1
      DY1(J) = (XB(J)) - YP(JJJ))/(YP(JJJ+1) - YP(JJJ))
  960 JYJ(J) = JJJ
      NU = 0
      DO 980 I=1,NPX
      DO 980 J=1,NPY
  980 JLT(J,I) = 2
      RETURN
C     BEAMS.
 1000 READ (5,2050) NPS
      WRITE (6,3400) NPS
      READ (5,2000) (SP(I),I=1,NPS)
      WRITE (6,3500) (SP(I),I=1NPS)
      WRITE (6,3300)
      READ (5,2050) (LTIME(I),I=1NPS)
      WRITE (6,2800) (LTIME(I),I=1,NPS)
      DO 1200 I=1,NPS
      NTIME = LTIME(I)
```

```
      READ(5,2000) (TTB(K,I),K=1,NTIME)
      WRITE (6,3700) I,(TTB(K,I)=1,NTIME)
      READ (5,2000) (PRTB(K,I),K=1,NTIME)
      WRITE (6,3800)
 1200 WRITE (6,3000) (PRTB(K,I),K=1,NTIME)
C     SPATIAL INTERPOLATION - EXTRAPOLATION.  INDICES ARE LOWER BOUND.
 1250 DO 1500 I=1,NPS
      FSP = SP(I)
      II = FSP + .00001
      DII = FSP - FLOAT(II)
      SPSX = 0.0
      IF (II.EQ.0) GO TO 1400
      DO 1300 J=1,II
 1300 SPSX = SPSX + DSOC(J)
 1400 IF (II.LT.NMASS+1) SPS(I) = SPSX + DII*DSOC(II+1)
 1500 CONTINUE
      DO 1600 J=2,NMASS
 1600 DSOO(J) = DSOO(J-1) + DSOC(J)
      DO 1850 I=1,NMASS
      DO 1700 III = 1,NPS
      IF (SPS(III).GT.DSOC(I)) GO TO 1800
 1700 CONTINUE
      III = NPS
 1800 IF (III.GT.1) III = III - 1
      DSOO(I) = (DSOO(I) - SPS(III))/(SPS(III+1) - SPS(III))
 1850 ISP(I) = III
      IF (NLOAD.EQ.4) GO TO 7100
      DO 1900 I=1,NPS
 1900 JLB(I) =2
      RETURN
C
C     LOAD OPTION 4 - TAPE INPUT FROM TAPE10.
C     PROGRAMMED FOR 7600 ONLY.
C
 6000 MAXD = 11914
      MAXD2 = 5957
      READ (5,2000) TIM1,SKIP
      TIM2 = TIM1 + TSTOP + DELTIM
      NSKIP = SKIP + .0001
      READ (5,2050) NGAGE
      READ (5,2050) (NORDER(I),I=1,NGAGE)
      WRITE (6,3900) TIM1,NSKIP,NGAGE,(NORDER(I),I=1,NGAGE)
      NGSUM = 0
      DO 6050 I=,NGAGE
      IF (NORDER(I).GT.0) NGSUM = NGSUM +1
 6050 CONTINUE
      IF (KTYPE.GT.5) GO TO 7000
C     PANELS.
      READ (5,2050) NPX,NPY
      WRITE (6,2700) NPX,NPY
```

```
      IF (NPX.GT.1.AND.NPY.GT.1) GO TO 8800
      READ (5,2000) (XP(I),I=1,NPX)
      READ (5,2000) (YP(I),I=1,NPY)
      WRITE (6,3100) (XP(I),I=1,NPX)
      WRITE (6,3200) (YP(I),I=1,NPY)
      JL = 2
      NU = 1
      IF (NPX*NPY.EQ.1) GO TO 7100
      NU = 0
      IF (NPX*NPY.NE.NGSUM) GO TO 8600
      IF (NPX.EQ.1) GO TO 6600
      DO 6500 I=1,NGT
      DO 6300 III=1,NPX
      IF (XP(III).GT.XG(I)) GO TO 6400
 6300 CONTINUE
      III = NPX
 6400 IF (III.GT.1) III = III - 1
      DX1(I) = (XG(I) - XP(III))/(XP(III+1) - XP(III))
 6500 IXI(I) = III
 6600 IF (NPY.EQ.1) GO TO 7100
      DO  6900 J=1,NBT
      DO 6700 JJJ = 1,NPY
      IF (YP(JJJ).GT.XB(J)) GO TO 6800
 6700 CONTINUE
      JJJ = NPY
 6800 IF (JJJ.GT.1) JJJ = JJJ - 1
      DUY1(J) = (XB(J) - YP(JJJ))/(YP(JJJ+1) - YP(JJJ))
 6900 JYJ(J) = JJJ
      GO TO 7100
C     BEAMS.
 7000 READ (5,2050) NPS
      WRITE (6,3400) NPS
      READ (5,2000) (SP(I) ,I=1,NPS)
      WRITE (6,3500) (SP(I),I=1,NPS)
      JL = 2
      NU = 1
      IF (NPS.EQ.1) GO TO 7100
      IF (NPS.NE.NGSUM) GO TO 8500
      NU = 0
      GO TO 1250
C
 7100 DEL = 0
      TCV = 1.E-6
      TIM1 = TIM1/TCV
      KG = 0
      KK = 0
      KKK = 0
      NTIME = 0
      BUFFER IN (10,1) (IC(1),IC(100)
      IF (UNIT(10)) 7200,7200,8300
```

```
 7200 BUFFER IN (10,1) (ID(1),ID(100))
      IF (UNIT(10)) 7250,8100,8300
 7250 IF (ID(1).EQ.TRD) GO TO 8700
      NWORDS = ID(17)
      NPOINT = ID(18)
      LR = NWORDS*NPOINT
      NWS = NWORDS*NSKIP
      KK = KK + 1
      KG = NORDER(KK)
      IF (KG.GT.0) KKK = KKK + 1
      IL = NWORDS
 7300 BUFFER IN (10,1) (AI(1),AI(LR))
      IF (UNIT (10)) 7400,8100,8300
 7400 IF (DEL.EQ.0.) DEL = (AI(1+NWORDS) - AI(1))*SPIP*TCV
C     LOCATE FIRST TIME.
       DO 7500 I=IL,LR,NWORDS
      IF (AI(I-1).GE.TIM1) GO TO 7600
 7500 CONTINUE
      GO TO 7300
 7600 IF (KG.GT.0) P(KG,1) = AI(I)
      T1 = AI(I-1)*TCV
      IF (NITME.EQ.0) NTIME = (TIM2-TIM1*TCV)/DEL + 2
      IF (NTIME.GT.MAXD) GO TO 8400
      J = 1
      IF (KG.EQ.0) GO TO 7800
      IL = NWS - LR + I
      IF (IL.GT.0) GO TO 7800
      IL = I + NWS
      DO 7700 I=IL,LR,NWS
      IX = I
      J = J + 1
      IF (J.GT.NITME) GO TO 7700
      P(KG,J) = AI(I)
 7700 CONTINUE
 7750 IL = NWS - LR + IX
 7800 BUFFER IN (10,1) (AI(1),AI(LR))
      IF (UNIT(10)) 7900,8100,8300
 7900 IF (KG.EQ.0) GO TO 7800
      IF (J.GE.NTIME) GO TO 7800
      DO 8000 I=IL,LR,NWS
      IX = I
      J = J + 1
      IF (J.GT.NTIME* GO TO 8000
      IF (J.GT.MAXD2) GO TO 7950
      P(KG,J) = AI(I)
      GO TO 8000
 7950 Q(KG,J-MAXD2) = AI(I)
 8000 CONTINUE
      GO TO 7750
C     END OF GAGE DATA.
```

```
 8100 IF (NDBUG.GT.0) WRITE (6,4200) KK,KG,DEL,T1,NTIME,
     1 NWORDS,NPOINT,(ID(I),I=1,9)
      IF (KKK.LT.NGSUM) GO TO 7200
C     DATA READ.
 8200 REWIND 10
      WRITE (6,4800) DEL,NTIME
      IF (NDBUG.LT.2) GO TO 8250
      DO 8220 I=1,NGSUM
      NT2 = NTIME - MAXD2
      WRITE (6,4900) I, (P(I,J)J=1,NT1)
      IF (NT2.GT.0) WRITE (6,4900) I,(Q(I,J),J=1,NT2)
 8220 CONTINUE
 8250 RETURN
C
C     ERRORS.
C
 8300 WRITE (6,4300)
      GO TO 8900
 8400 WRITE (6,4400) NTIME,MAXD
      GO TO 8900
 8500 WRITE (6,4500) NPS,NGSUM
      GO TO 8900
 8600 WRITE (6,4500) NPX,NPY,NGSUM
      GO TO 8900
 8700 WRITE (6,4700) KK,NGAGE
      GO TO 8900
 8800 WRITE (6,4000)
C
 8900 KERR = 2
      RETURN
C
 2000 FORMAT(6F12.1)
 2050 FORMAT (6I12)
 2100 FORMAT 2F12.1)
 2200 FORMAT (24HOSTATIC PRESSURE, PSI = E15.6)
 2300 FORMAT (23HODYNAMIC LOAD CONSTANTS/
     1       11H  PP1    = E15.6/
     1       11H  PPO    = E15.6/
     1       11H  TTO    = E15.6/
     1       11H  TPRIME = E15.6/
     1       11H   AA    = E15.6/
     1       11H   ANN   = E15.6)
 2400 FORMAT (21HODYNAMIC LOAD OPTION I4)
 2500 FORMAT (18HONUMBER OF TIMES = I4/28H   TIME, SEC  PRESSURE, PSI/
     1  (2E15.6))
 2700 FORMAT (24HONUMBER OF LOAD STATIONS/
     1  12H   NPX   = I3/12H   NPY   = I3)
 2800 FORMAT (5X,10I5)
 2900 FORMAT (18H PRESSURES (PSI) =)
 3000 FORMAT (5X,6E15.6)
 3100 FORMAT (19HOX-POSITIONS (IN) =/(5X,5E15.6))
```

```
3200 FORMAT (26HOY-POSITIONS (IN) OR DEG) =/5X,5E15.6))
3300 FORMAT (26HONUMBER OF TABLE ENTRIES =)
3400 FORMAT (24HONUMBER OF LOAD STATIONS/12H   NPS   = I3)
3500 FORMAT (25HOMEASUREMENT POSITIONS = /5X,5E15.6))
3600 FORMAT (14HOTIMES (SEC)  ,10X,6HNPX = 20,5X6HNPY = I3/
    1  (5X,6E15.6))
3700 FORMAT (14HOTIMES (SEC)   ,10X,2HI=I3/(5X,6E15.6))
3800 FORMAT (19H PRESSURES (SEC) = )
3900 FORMAT (9HOTAPE USE/ 34H  START TIME, SEC (TIM1)     = E15.6/
    1  34H   SKIP  FREQUENCY (NSKIP)      = I6/
    2  34H   NO. OF GAGES ON TAPE (NGAGE) = I6/
    3 25H   LOCATION ID OF GAGES = /(5X,10I4))
4000 FORMAT (33HOTAPE INPUT IS ONLY 1 DIMENSIONAL/
    1  28H EITHER NPX OR NPY MUST BE 1)
4200 FORMAT (19HODATA FOR GAGE NO. I4, 15H,  LOCATION ID  I4/
    1  14H TIME INTERVAL E15.6, 13H,  START TIME E15.6/
    2  16H NUMBER OF TIMES I6/ 26H NUMBER OF WORDS PER POINT I4/
    3  28H NUMBER OF POINST PER RECORD I5/ 1X,9A10)
4300 FORMAT (26HOPARITY ERROR ON DAT TAPE)
4400 FORMAT (25HODATA EXCEEDS TABLE SPACE 2I5)
4500 FORMAT (32HONUMBER OF ACTIVE GAGES IS WRONG 3I4)
4700 FORMAT (12HO END OF TAPE 2I4)
    1  19H   TIME INTERVAL = E15.6/18H  NO. OF TIMES = I6)
4900 FORMAT (5HO I = I3,4X,9HP (PSI) = /(1X,10E12.4))
     END
*PURDECK PRESS
*PURGE PRESS
*ADDFILE INPUT,INT1
*DECK PRESS
      SUBROUTINE PRESS
*CALL CNOVA
*CALL CLOAD
*CALL CBLK1
*CALL COM1
*CALL COM2
      DIMENSION PRTTB(41)
      DIMENSION TTB(14,41),PRTB(14,41)
      EQUIVALENCE (PRT(1,1,1),PRTB(1,1)), (TTP(1,1,1),TTB(1,1))
      EQUIVALENCE (PRTT(1,1),PRTTB(1))
C
      IF(NCALL.GT.0) GO TO 9000
      ZZ= 1.0/RTRIAL(1)
      GO TO (50,220,800,1000), NLOAD
  50 IF(TIME.GE.TPRIME) TO TO 100
      PPP=ZZ*PP1*(1.0 - TIME*OTT1)
      IF(PPP.LT.0.0) PPP=0.0
      GO TO 400
 100  IF(TIME.GE.TTO) GO TO 200
      PPP=PPO*(1.0 - TIME*OTTO)**ANN
      PPP=ZZ*PPP*EXP(-AZ*TIME)
```

```
      GO TO 400
  200 PPP=0.0
      GO TO 400
C
  220 DO 240 J=JL,NTIME
      IF (TIME.LE.TT(J)) GO TO 260
  240 CONTINUE
      JL = NTIME
      PPP = ZZ*PT(JL)
      GO TO 400
  260 JL = J
      PPP = PT(J-1) + (TIME - TT(J-1))*(PT(J) - PT(J-1))/
     1  (TT(J) - TT(J-1))
      PPP = ZZ*PPP
C
  400 IF (KTYPE.LT.6) GO TO 9000
      PX = PPP
      IF (KTYPE.EQ.10) PX = 0.
      DO 500 I=1,NMASS
  500 PB(I) = PX
      GO TO 9000
C
  800 IF (KTYPE.GT.5) GO TO 900
C     PANELS.
      DO 860 I=1,NPX
      DO 860 J=1,NPY
C     INTERPOLATE ON TIME.
      PPP = 0.0
      IF (TIME.LT.TTP(1,J,I)) GO TO 860
      JL = JLT(J,I)
      NTIME = KTIME(J,I)
      DO 820 K=JL,NTIME
      KK = K
      IF (TIME.LE.TTP(K,J,I)) GO TO 840
  820 CONTINUE
      JLT(J,T) = NTIME
      PPP=PRT(NTIME,J,I)
  840 JL = KK
      P1 = PRT(JL-1,J,I)
      T1 = TTP(JL-1,J,I)
      PPP = P1 + (TIME - T1)*(PRT(JL,J,I) - P1)/(TTP(JL,J,I) - T1)
      JLT(J,I) = JL
  860 PRTT(J,I) = PPP
C     INTERPOLATE SPATIALLY.
      K = 0
      DO 880 I=1,NGT
      II = IXI(I)
      DX = DX1(I)
      DO 880 J=1,NRT
      IF (NUSE(J,I).EQ.0) GO TO 880
```

```
      K = K + 1
      JJ = JYJ(J)
      DY = DY1(J)
      P1 = PRTT(JJ,II) + DY*(PRTT(JJ+1,II) - PRTT(JJ,II))
      P2 = PRTT(JJ,II+1) + DY*(PRTT(JJ+1,II+1) - PRTT(JJ,II+1))
      PPP = P1 + DX*(P2 - P1)
      PA(K) = PPP*ZZ
  880 CONTINUE
      GO TO 9000
C     BEAMS.
  900 DO 930 I=1,NPS
      PPP = 0.0
      IF (TIME.LT.TTB(1,I)) GO TO 930
      JL = JLB(I)
      NTIME = LTIME(I)
      DO 910 K=JL,NTIME
      KK = K
      IF (TIME.LE.TTB(K,I)) GO TO 920
  910 CONTINUE
      JLB(I) = NTIME
      PPP = PRTB(NTIME,I)
      GO TO 930
  920 JL = KK
      P1 = PRTB(JL-1,I)
      T1 = TTB(JL-1,I)
      PPP = P1 + (TIME-T1)*(PRTB(JL,I) - P1/(TTB(JL,I) - T1)
      JLB(I) = JL
  930 PRTTB(I) = PPP
      K = 0
      DO 940 I=1,NMASS
      II = ISP(I)
      DX = DSOC(I)
      PPP = PRTTB(II) +DX*(PRTTB(II+1) - PRTTB(II))
  940 PB(I) = PPP*ZZ
      GO TO 9000
C
C     TAPE OPTION.
C
 1000 IF (NU.EQ.0) GO TO 1500
C     UNIFORM LOAD.
      DO 1100 K=JL,NTIME
      T1 = DEL*FLOAT(K-1)
      IF (TIME.LE.T1) GO TO 1200
 1100 CONTINUE
      JL = NTIME
      IF (NTIME.LE.MAXD2) PPP = P(1,NTIME)
      IF (NTIME.GT.MAXD2) PPP = Q(1,NTIME-MAXD2)
      GO TO 1300
 1200 JL = K
      T1 = T1 - DEL
```

```
      T1 = (TIME-T1)/DEL
      IF (JL-1.GT.MAXD2) GO TO 1220
      IF (JL.GT.MAXD2) GO TO 1230
      P2 = P(1,JL)
      GO TO 1250
 1220 P1 = Q(1,JL-1-MAXD2)
 1230 P2 = Q(1,JL-MAXD2)
 1250 PPP = P1 + T1*(P2-P1)
 1300 PPP = PPP*ZZ
      IF (KTYPE.LT.6) GO TO 9000
      PX = PPP
      IF (KTYPE.EQ.10) PX = 0.
      DO 1400 I=1,NMASS
 1400 PR(I) = PX
      GO TO 9000
C     NON-UNIFORM LOAD.
 1500 DO 1600 K=JL,NTIME
      T1 = DEL*FLOAT(K-1)
      IF (TIME.LE.T1) GO TO 1700
 1600 CONTINUE
      JL = NTIME
      GO TO 1800
 1700 JL = K
      T1 = T1 -DEL
      T1 = (TIME-T1)/DEL
 1800 IF (KTYPE.GT.5) GO TO 2400
C     PANELS.
      IF (JL.LT.NTIME) GO TO 1900
      DO 1850 KG= 1,NGSUM
      IF (NTIME.LE.MAXD2) PRTTB(KG) = P(KG,NTIME)
      IF (NTIME.GT.MADX2) PRTTB(KG) = Q(KG,NTIME-MAXD2)
 1850 CONTINUE
      GO TO 2000
 1900 DO 1950 KG=1,NGSUM
      IF (JL-1.GT.MAXD2) GO TO 1920
      P1 = P(KG,JL-1)
      IF (JL.GT.MAXD2) GO TO 1930
      P2 = P(KG,JL)
      GO TO 1950
 1920 P1 = Q(KG,JL-1-MAXD2)
 1930 P2 = Q(KG,JL-MAXD2)
 1950 PRTTB(KG) = P1 + T1*(P2-P1)
C
 2000 K = 0
      DO 2300 I=1,NGT
      IF (NPX.EQ.1) GO TO 2050
      II = IXI(I)
      DX = DX1(I)
 2050 DO 2300 J=1,NBT
      IF (NUSE(J,I).EQ.0) GO TO 2300
```

```
      K = K + 1
      IF (NPY.EQ.1) GO TO 2100
      JJ = JYJ(J)
      DY= DY1(J)
      P1 = PRTTB(JJ)
      PPP = P1 + DY*(PRTTB(JJ+1) - P1)
      PA(K) = PPP*ZZ
      GO TO 2300
 2100 IF (J.GT.1) GO TO 2200
      P1=PRTTB(II)
      PPP = (P1 + DX*(PRTTB(II+1) - P1))*ZZ
 2200 PA(K) = PPP
 2300 CONTINUE
      GO TO 9000
C     BEAMS.
 2400 IF (JL.LT.NTIME) GO TO 2500
      DO 2450 KG=1,NGSUM
      IF (NTIME.GT.MAXD2) PRTTB(KG) = Q(KG,NTIME-MAXD2)
 2450 CONTINUE
      GO TO 2600
 2500 DO 2550 KG=1,NGSUM
      IF (JL-1.GT.MAXD2) GO TO 2520
      P1 = P(KG,JL-1)
      IF (JL.GT.MAXD2) GO TO 2530
      P2 = P(KG,JL)
      GO TO 2550
 2520 P1 = Q(KG,JL-1-MAXD2)
 2530 P2 = Q(KG,JL-MAXD2)
 2550 PRTTB(KG) = P1 + T1*(P2-P1)
 2600 DO 2700 I=1,NMASS
      II = ISP(I)
      DX = DSOC(I)
      P1 = PRTTB(II)
 2700 PR(I) = ZZ*(P1 + DX*(PRTTB(II+1) - P1))
C
 9000 RETURN
      END
```

APPENDIX B

RIB BUCKLING OPTION CHANGES

This appendix contains the changes to the NOVA-2LT (NOVA-2LTS) codes which are necessary to include pinned/sliding-pinned end constraints for the rib buckling option (KTYPE=10).

Since the NOVA-2 and NOVA-2S program libraries are maintained at the AFWL using the Control Data Corporation 6000/7000 UPDATE program, the UPDATE directives needed to accomplish these changes are also included.

CHANGES TO EXTEND
NOVA2-LT FOR EXPERIMENTAL BUCKLING

```
*IDENT AUG0478

*INSERT COMP2.51
      IF (KEYB2.EQ.2) GO TO 3970
      TH = ATAN2(W(1)-W1,V(1)-V1)
      STH2 = SIN(TH)
      CTH2 = COS(TH)
*INSERT COMP2.103
      IF (KEYB1*KTYPE.EQ.30) IL1(I) = 2
*INSERT COMP2.114
      IF (KEYB1.EQ.3) EL = PI/SQRT((VM(1)-V2)**2 + (WM(1)-W2)**2)
*INSERT COMP2.116
      IF (KEYB2.EQ.3) IL = N
*INSERT COMP2.119
      IF (KEYB1.EQ.2) GO TO 4075
      D = SQRT((VM(I) - VM(1))**2 + (WM(I) - WM(1))**2)
      D = AMP*SIN(D*EL)
 4075 CONTINUE
*INSERT COMP2.137
      SM(1) = .00298/DELTS(1)
*INSERT CYCLE.7
      IF (KTYPE.EQ.10) PPP = PPP*WT*WR(1)
*DELETE CYCLE.9
*INSERT CYCLE.52
      IF (KTYPE.EQ.10) CRIT(1) = -1.0
*DELETE EQUILP.54
  250 CONTINUE
*INSERT EQUILP.61
      IF (KEYB1*KTYPE.NE.30) RETURN
C     PINNED-SLIDING CONSTRAINT.
      ACCNV(1)=C6(1)*(PPP*COST(1)+BIGN(2)*COST(2)-Q(2)*SINT(2))
      ACCNW(1) = 0.0
*INSERT FB.37
      KTYPEX = KTYPE
      KTYPE = 7
*INSERT FB.230
      KTYPE = KTYPEX
*INSERT FB.236
      KTYPE = KTYPEX
*INSERT FB.242
      KTYPE = KTYPEX
```

CHANGES TO EXTEND
NOVA-2LTS FOR EXPERIMENTAL BUCKLING

```
*IDENT AUG0378

*INSERT COMP2.51
      IF (KEYB2.EQ.2) GO TO 3970
      TH = ATAN2(W(1)-W1,V(1)-V1)
      STH2 = SIN(TH)
      CTH2 = COS(TH)
*INSERT COMP2.103
      IF (KEYB1*KTYPE.EQ.30) IL1(I) = 2
*INSERT COMP2.114
      IF (KEYB1.EQ.3) EL = PI/SQRT((VM(1)-V2)**2 +(WM(1)-W2)**2)
*INSERT COMP2.116
      IF (KEYB2.EQ.3) IL = N
*INSERT COMP2.120
      IF (KEYB1.EQ.2) GO TO 4075
      D = SQRT((VM(I) - VM(1))**2 + (WM(I) - WM(1))**2)
      D = AMP*SIN(D*EL)
 4075 CONTINUE
*INSERT COMP2.138
      SM(1) = .00339/DELTS(1)
*INSERT CYCLE.7
      IF (KTYPE.EQ.10) PPP = PPP*WT*WR(1)
*DELETE CYCLE.9
*INSERT CYCLE.52
      IF (KTYPE.EQ.10) CRIT(1) = -1.0
*INSERT EQUILP.61
      IF (KEYB1*KTYPE.NE.30) RETURN
C     PINNED-SLIDING CONSTRAINT.
      ACCNV(1)=C6(1)*(PPP*COST(1)+BIGN(2)*COST(2)-Q(2)*SINT(2))
      ACCNW(1) = 0.0
*INSERT FB.37
      KTYPEX = KTYPE
      KTYPE = 7
*INSERT FB.230
      KTYPE = KTYPEX
*INSERT FB.236
      KTYPE = KTYPEX
```

APPENDIX C

FREE BOUNDARY CONDITIONS AND DISCRETE LINEAR SPRINGS OPTION CHANGES

This appendix contains the changes to the NOVA-2LTS code which are necessary to include free boundary conditions and discrete linear springs.

Since the NOVA-2S program library is maintained at the AFWL using the Control Data Corporation 6000/7000 UPDATE program, the UPDATE directives needed to accomplish these changes are also included.

EXTEND CEPROP TO INCLUDE
F-F AND C-F EDGE CONDITIONS AND ELASTIC SPRINGS

```
*IDENT OCT0378

*DELETE JUNE2678.3
     3   PI,XB(28),XG(28),NBND1,NBND2,NSPR,IDIR(30),NSPG(30),
     4   NSPB(30),BIGK(30),XLP3
*DELETE BOLT.9
      DIMENSION CD1(20),CD2(20),CC3(20),CD4(20),CD5(20),CD6(20)
*INSERT BOLT.20
      DATA CD5/0.596864162698, 1.49417561426, 2.50024694616,
     1   3.49998931984, 4.50000046151, 5.49999998001, 6.50000000087,
     6   7.49999999995, 8.5,9.5,10.5,11.5,12.5,13.5,14.5,15.5,16.5,
     3   17.5,18.5,19.5/
      DATA CD6/0.734095513769, 1.01846731877, 0.999224496517,
     1   1.00003355325, 0.999998550107, 1.00000006264, 0.999999997294,
     2   1.00000000011, 12*1.0/
*INSERT BOLT.23
      FAC1 = SQRT(3.0)
*DELETE BOLT.27
      GO TO (500,700,930,900,960), NBND1
*INSERT BOLT.65
C
C     FREE-FREE, GAMMA.
C
  930 DO 950 I=1,MG
      M = MGM(I)
      X1 = CD1(M)
      X2 = CD2(M)
      DO 950 J=1,NGT
      II = II + 1
      IF (M-2) 935,940,945
 935  FP1(II) = 1.0
      FP2(II) = 0.
      FP3(II) = 0.
      GO TO 950
  940 FP1(II) = FAC1*(1.0 - 2.0*GAM(J)/PI)
      FP2(II) = -FAC1*2.0/PI
      FP3(II) = 0.
      GO TO 950
  945 X3 = X1*GAM(J)
      EX1 = EXP(X3)
      EX2 = EXP(-X3)
      SL = SIN(X3)
      CL = COS(X3)
      FP1(II) = CL - X2*SL + .5*(1.-X2)*EX1 + .5*(1. + X2)*EX2
      FP2(II) = X1*(-SL - X2*CL + .5*(1.-X2)*EX1 - .5*(1.+X2)*EX2)
      FP3(II) = X1**2*(-CL + X2*SL + .5*(1.-X2)*EX1 + .5*(1.+X2)*EX2)
  950 CONTINUE
      CK(5) = 1./FAC
      GO TO 1000
C
```

```
C      CLAMPED-FREE, GAMMA.
C
  960 DO 980 I=1,MG
      M = MGM(I)
      CDL(I) = CD5(M)
  980 CDA(I) = CD6(M)
      GO TO 540
*DELETE BOLT.76
      GO TO (1100,1300,1530,1500,1580), NBND2
*INSERT BOLT.121
C
C      FREE-FREE, BETA.
C
 1530 DO 1570 I=1,MB
      N = NBN(I)
      X1 = CD1(N)
      X2 = CD2(N)
      DO 1570 J=1,NB1
      II = II + 1
      IF (N-2) 1540,1550,1560
 1540 FP5(II) = 1.0
      FP6(II) = 0.
      FP7(II) = 0.
      GO TO 1570
 1550 FP5(II) = FAC1*(1. -2.*BETR(J)/P1)
      FP6(II) = -FAC1*2.0/PI
      FP7(II) = 0.
      GO TO 1570
 1560 X3 = X1*BETR(J)
      EX1 = EXP(X3)
      EX2 = EXP(-X3)
      SL = SIN(X3)
      CL = COS(X3)
      FP5(II) = CL - X2*SL + .5*(1.-X2)*EX1 + .5*(1. + X2)*EX2
      FP6(II) = X1*(-SL - X2*CL + .5*(1.-X2)*EX1 - .5*(1.+X2)*EX2)
      FP7(II) = X1**2*(-CL + X2*SL + .5*(1.-X2)*EX1 + .5*(1.+X2)*EX2)
 1570 CONTINUE
      CK(6) = 1.0/FAC
      GO TO 1600
C
C      CLAMPED-FREE, BETA.
C
 1580 DO 1590 I=1,MB
      N = NBN(I)
      CDL(I) = CD5(N)
 1590 CDA(I) = CD6(N)
      GO TO 1140
*DELETE DERV2.245
C
C      ELASTIC SPRINGS.
C
      IF (NSPR.EQ.0) GO TO 1740
      DO 1730 L12 = 1,NSPR
```

```
      L1 = NSPG(L12)
      L2 = NSPB(L12)
      K = (L1-1)*NBT + L2
      BIGKL = BIGK(L12)*XLP3
      IF (IDIR(L12) - 2) 1710,1720,1725
 1710 SURS = BIGKL*U(K)*SIN G(MMC+L1)*SIN2B(NNO+L2) + SURS
      GO TO 1730
 1720 SVRS = BIGKL*V(K)*SIN2G(MMC/+L1)*SINB(NNO+L2) + SVRS
      GO TO 1730
 1725 SWRS = BIGKL*W(K)*FP1(MMC+L1)*FP5(NNO+L2) + SWRS
 1730 CONTINUE
 1740 IF (ABS(SWRS).GT.1.E20) GO TO 2150
*DELETE DSET1.53
  190 READ (5,7000) NSPR
      IF (NSPR.EQ.0) GO TO 210
      DO 200 I=1,NSPR
  200 READ (5,7110) IDIR(I),NSPG(I),NSPB(I),BIGK(I)
  210 KSTIF = IABS(NSG) + IABS(NSB)
*DELETE DSET1.164
 1190 WRITE (6,12000) NSPR
      IF  (NSPR.EQ.0) GO TO 1210
      WRITE (6,12200)
      DO 1200 I=1,NSPR
 1200 WRITE (6,12100) IDIR(I),NSPG(I),NSPB(I),BIGK(I)
 1210 IF  (NSG.EQ.0) GO TO 1300
*INSERT DSET1.219
      NBND1 = NBND/10
*INSERT DSET1.223
 7110 FORMAT (3I12,F12.1)
*DELETE DSET1.225
 7170 FORMAT (75H1INPUT DATA FOR CEPROP (MODIFIED TO INCLUDE EXTRA B.C.
     1AND ELASTIC SPRINGS))
*INSERT DSET1.276
12000 FORMAT (8H0NSPR = I3)
12100 FORMAT (5X,3I6,E15.6)
12200 FORMAT (37H        IDIR    NSPG    NSPR     K (LB/IN))
*DELETE DSET3.32,DSET3.37
 2950 IF (NBND1.EQ.1) WRITE (6,9900)
      IF (NBND1.EQ.2) WRITE (6,9920)
      IF (NBND1.EQ.3) WRITE (6,9930)
      IF (NBND1.EQ.4) WRITE (6,9940)
      IF (NBND1.EQ.5) WRITE (6,9950)
      IF (NBND2.EQ.1) WRITE (6,9960)
      IF (NBND2.EQ.2) WRITE (6,9980)
      IF (NBND2.EQ.3) WRITE (6,9990)
      IF (NBND2.EQ.4) WRITE (6,10000)
      IF (NBND2.EQ.5) WRITE (6,10050)
*INSERT DESET3.73
      WRITE (6,13400) NSPR
*DELETE DSET3.88,DSET3.89
      IF (NBND1.GT.3) NNSYMG = 1
      IF (NBND2.GT.3) NNSYMB = 1
*INSERT DSET3.126
```

```
      IF (NSPR.EQ.0) GO TO 4075
      XLP3 = 2.0*XJ/(A*PI*H*XLP)
      IF (NSYMG*NSYNB.EQ.1) GO TO 4075
C
C     ACCOUNT FOR SYMMETRY.
C
      XLP3 = 4.0*XLP3/ FLOAT(NSYMG+1)*(MSYNB+1))
      DO 4060 L=1,NSPR
      I = NSPG(L)
      J = NSPB(L)
      IF(I.EQ.MBAR) BIGK(L) = 0.5*BIGK(L)*FLOAT(NSYMG+1)
      IF (J.EQ.NBAR) BIGK(L) = 0.5*BIGK(L)*FLOAT(NSYMB+1)
 4060 CONTINUE
 4075 CONTINUE
*DELETE MAY0878.31
 9300 FORMAT (1H1,20X,33HD E P R O P (EXTENDED OCT., 1978)/
     1  15H0PANEL ANALYZED)
*INSERT DSET3.231
 9930 FORMAT (31H   FREE - FREE, GAMMA DIRECTION)
*INSERT DSET3.232
 9950 FORMAT (34H   CLAMPED - FREE, GAMMA DIRECTION)
*INSERT DSET3.234
 9990 FORMAT (30H   FREE - FREE, BETA DIRECTION)
*INSERT DSET3.235
10050 FORMAT (33H   CLAMPED - FREE, BETA DIRECTION)
*INSERT DSET3.283
13400 FORMAT (38H0NUMBER OF DISCRETE ELASTIC SPRINGS = I3)
*DELETE DTSTEP.19,DTSTEP.20
       IF (NBND1.EQ.4) CM = .15
       IF (NBND1.EQ.1) CM = .3
*DELETE DTSTEP.22,DTSTEP.23
       IF (NBND2.EQ.4) CN = .15
       IF (NBND2.EQ.1) CN = .3
*IDENT AUG1678
*DELETE BLK6.6
     4    STP1(10,2),STRA(5,21,20),ST1,ST2,TEND1,TEND2,TETA(11),
*ID OCT0278
*I PINIT.54
       NU=1
```

DISTRIBUTION

DTIC/DDA
DNA/WashDC
NWEF/KAFB
ASD/WPAFB
AFWL/HO
AFWL/SUL
AFWL/NTYV
AFWL/NTYV/Off Rec Cy
AUL/LDE
Boeing/Seattle
Boeing/Wichita
Effects Technology/Santa Barbara
Kaman AviDyne/Burlington
McDonnell-Douglas/Long Beach
AFIT/WPAFB
R&D Assoc/Marina del Rey

DAT
FILM

0-